



POLITECHNIKA WARSZAWSKA
WYDZIAŁ MATEMATYKI I NAUK INFORMACYJNYCH



PRACA DYPLOMOWA MAGISTERSKA
MATEMATYKA

**METODY ŁĄCZENIA KLASYFIKATORÓW
W ANALIZIE DYSKRYMINACYJNEJ**

AUTOR:
IWONA GŁOWACKA

PROMOTOR:
PROF. DR HAB. JACEK KORONACKI

WARSZAWA, WRZESIEŃ 2006

.....

Podpis promotora

.....

Podpis autora

*Serdeczne podziękowania
Profesorowi Jackowi Koronackiemu
oraz Mariuszowi Gromadzie
za bardzo cenną pomoc w trakcie
pisanie niniejszej pracy*

Iwona Głowacka

Spis treści

Wstęp	7
1. Analiza dyskryminacyjna	9
1.1. Wprowadzenie	9
1.2. Zadanie klasyfikacji pod nadzorem	9
1.3. Liniowa funkcja dyskryminacyjna	13
1.3.1. Problem dwóch klas	13
1.4. Dyskryminacja oparta na rozkładach prawdopodobieństwa	16
1.4.1. Reguła największej wiarygodności	17
1.4.2. Reguła Bayesa	18
1.5. Dyskryminacja jako problem regresji	20
1.6. Dyskryminacja logistyczna	21
1.7. Ocena jakości klasyfikacji	22
1.7.1. Błąd klasyfikacji	22
1.7.2. Proces uczenia oraz predykcji	25
2. Rodziny klasyfikatorów	27
2.1. Wprowadzenie	27
2.2. Pojęcie rodziny klasyfikatorów	27
2.3. Dlaczego rodziny klasyfikatorów?	28
2.4. Nota historyczna	29
3. Metoda bagging	31
3.1. Wprowadzenie	31
3.2. Bootstrap i rodzina bagging	31
3.3. Dlaczego bagging działa?	32
3.3.1. Teoretyczna definicja baggingu	33
4. Metoda boosting	36
4.1. Wprowadzenie	36
4.2. Algorytm AdaBoost	36
4.3. AdaBoost jako addytywny model regresji logistycznej	37
4.4. Błąd klasyfikacji algorytmem AdaBoost	40
4.4.1. Asymptotyczna optymalność algorytmu AdaBoost	41
4.5. Modyfikacje algorytmu AdaBoost	43

5. Lasy losowe	44
5.1. Wprowadzenie	44
5.2. Drzewa klasyfikacyjne	44
5.2.1. Struktura drzewa	44
5.2.2. Kryteria podziałów	45
5.2.3. Kryteria jakości podziałów	46
5.2.4. Przycinanie drzew	47
5.3. Metoda lasów losowych	48
5.4. Moc i korelacja	50
6. Analiza danych	52
6.1. Wprowadzenie	52
6.2. Granica podziału	53
6.3. Błąd klasyfikacji	53
6.3.1. Analiza danych BreastCancer	54
6.3.2. Analiza danych Vowel	54
6.3.3. Analiza pozostałych danych	55
6.4. Podsumowanie	56

Wstęp

Temat pracy dotyczy problemu dyskryminacji oraz budowy i zastosowań rodzin klasyfikatorów, w tym głównie metody typu „bagging”, metody typu „boosting” oraz lasów losowych. Przedmiotem pracy jest zbadanie metematemyczno-statystycznych fundamentów, na których opierają się metodologie budowy rodzin klasyfikatorów. Istotną częścią pracy jest analiza rozwiązań podanych zagadnień.

W pierwszym rozdziale omówiony został problem klasyfikacji pod nadzorem, zwanej analizą dyskryminacyjną. Podano model analizy dyskryminacyjnej oraz przedstawiono podstawowe metody rozwiązań podanych zagadnień. Dużo uwagi poświęcono ocenie jakości klasyfikacji.

Rozdział drugi skupia się na idei łączenia klasyfikatorów, w tym przede wszystkim na podaniu i uzasadnieniu ich zalet. Wprowadzono precyzyjną definicję rodziny oraz miarę pewności predykcji opartej na rodzinie klasyfikatorów.

Kolejne trzy rozdziały poświęcone są wspomnianym metodom łączenia klasyfikatorów w analizie dyskryminacyjnej. Rozdział trzeci omawia metodę typu „bagging”. Rozdział czwarty przedstawia metodę typu „boosting”. Natomiast rozdział piąty skupia się na metodzie lasów losowych.

Pracę kończy szeroka analiza danych, potwierdzająca własności rozważanych metod.

Rozdział 1

Analiza dyskryminacyjna

1.1. Wprowadzenie

W efekcie ogromnego wzrostu mocy obliczeniowej komputerów oraz możliwości przechowywania i przetwarzania dużych ilości danych, komputery stały się integralną częścią naszego świata. Jesteśmy świadkami prawdziwej eksplozji baz danych, mając na myśli ich liczbę i objętość. Prostota konstrukcji oraz akceptowalny koszt sprawiły, że systemy gromadzące dane¹ stosuje się prawie we wszystkich dziedzinach życia. Jednym z głównych celów gromadzenia danych jest odkrywanie ukrytych w nich zależności². Nowoczesnych metod analizy danych dostarcza współczesna statystyka wielowymiarowa, gdzie szczególnie praktyczne znaczenie mają *metody klasyfikacji*. Rozróżniamy dwa typy klasyfikacji. Pierwszy to *klasyfikacja pod nadzorem*, określana mianem *analizy dyskryminacyjnej* lub *klasyfikacją z próbą uczącą*. Drugi typ to *klasyfikacja bez nadzoru* nazywana *analizą skupień*. W poniższym tekście przedstawimy zadanie klasyfikacji pod nadzorem.

1.2. Zadanie klasyfikacji pod nadzorem

Rozważmy g różnych (rozłącznych) populacji, $g \geq 2$ w pewnej populacji generalnej oraz p -wymiarową próbę losową $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n)$, gdzie każdy z wektorów losowych ma postać

$$\mathbf{X}_i = (X_i^{(1)}, X_i^{(2)}, \dots, X_i^{(p)}) \quad i = 1, 2, \dots, n \quad (1.1)$$

oraz $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(p)}) \in \mathbb{X}$ jest wartością i -tego wektora losowego. Ustalmy realizację $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ próby \mathbf{X} . Realizację \mathbf{x} można zapisać w poniższej postaci:

$$\begin{aligned} \mathbf{x}_1 &= (x_1^{(1)}, x_1^{(2)}, \dots, x_1^{(p)}) \\ \mathbf{x}_2 &= (x_2^{(1)}, x_2^{(2)}, \dots, x_2^{(p)}) \\ &\vdots \\ \mathbf{x}_n &= (x_n^{(1)}, x_n^{(2)}, \dots, x_n^{(p)}) \end{aligned} \quad (1.2)$$

Elementy $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{X}$ realizacji \mathbf{x} nazywamy *obserwacjami*. Zbiór \mathbb{X} nazywamy *zbiorem obserwacji*. Wspólny rozkład zmiennych losowych $X_i^{(k)}$ ($i = 1, 2, \dots, n$) jest rozkładem

¹Systemy gromadzące dane to najczęściej hurtownie danych (z ang. data warehouse).

²Odkrywanie zależności w danych nazywa się również eksploracją danych (z ang. data mining).

pewnej cechy $X^{(k)}$. Cechę $X^{(k)}$ nazywamy k -tym *atrybutem*. Atrybuty mogą być mierzone na dowolnej skali, czyli mogą to być zmienne *ciągłe*, *dyskretne*, *porządkowe* lub *nominalne*³. Przykładem atrybutu ciągłego jest wzrost wyrażony w jednostkach miary długości. Liczba posiadanych dzieci to atrybut dyskretny. Wzrost wysoki, średni lub niski jest atrybutem porządkowym. W przypadku nazw kolorów (zielony, żółty) do czynienia mamy z atrybutem nominalnym.

Rozważane populacje nazywamy również *klasami* i czasami *grupami*. Klasy kodujemy liczbami ze zbioru $\{1, 2, \dots, g\}$. Wyjątek od tej reguły spotykamy jedynie w sekcjach 1.6 i 4.3. Każda obserwacja \mathbf{x}_i pochodzi więc z pewnej klasy o etykiecie $y_i \in \{1, 2, \dots, g\}$. Zbiór dostępnych danych wygodnie będzie zapisać jako ciąg n uporządkowanych par losowych

$$\mathcal{L} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\} \quad (1.3)$$

lub krótko

$$\mathcal{L} = \{(\mathbf{x}_i, y_i), \quad i = 1, 2, \dots, n\} \quad (1.4)$$

gdzie $\mathbf{x}_i \in \mathbb{X}$ oznacza i -tą obserwację, natomiast $y_i \in \{1, 2, \dots, g\}$ jest etykietą klasy, do której ta obserwacja należy. Ogólnie pisząc (\mathbf{x}, y) myślimy o obserwacji \mathbf{x} z klasy o etykiecie y .

płeć	wiek	BMI	obwód pasa	nadciśnienie tętnicze	zawał serca
K	55	27.2	75	umiarkowane	tak
K	42	22.9	62	brak	nie
M	31	33.1	110	umiarkowane	tak
K	68	26.3	76	ciężkie	tak
M	72	29.6	98	umiarkowane	nie
M	43	27.9	90	brak	nie
M	36	24.7	89	brak	nie
K	61	30.2	84	ciężkie	tak
K	84	31.0	86	ciężkie	tak
M	25	20.2	70	brak	nie
K	33	19.5	55	brak	tak
M	47	27.7	96	brak	tak
K	52	29.8	77	łagodne	nie

Tabela 1.1: Przykładowy zbiór danych.

Przykład 1.2.1 W tabeli 1.1 przedstawiono dane o pacjentach pewnej placówki medycznej. Dane zawierają 5 atrybutów: płeć (nominalny, „K” - kobieta, „M” - mężczyzna), wiek (dyskretny, liczba lat), BMI⁴ - współczynnik masy ciała (ciągły, $[kg/m^2]$), obwód pasa (ciągły, $[cm]$), nadciśnienie tętnicze⁵ (porządkowy, kategoryzacja). Zmienna „zawał serca” dostarcza

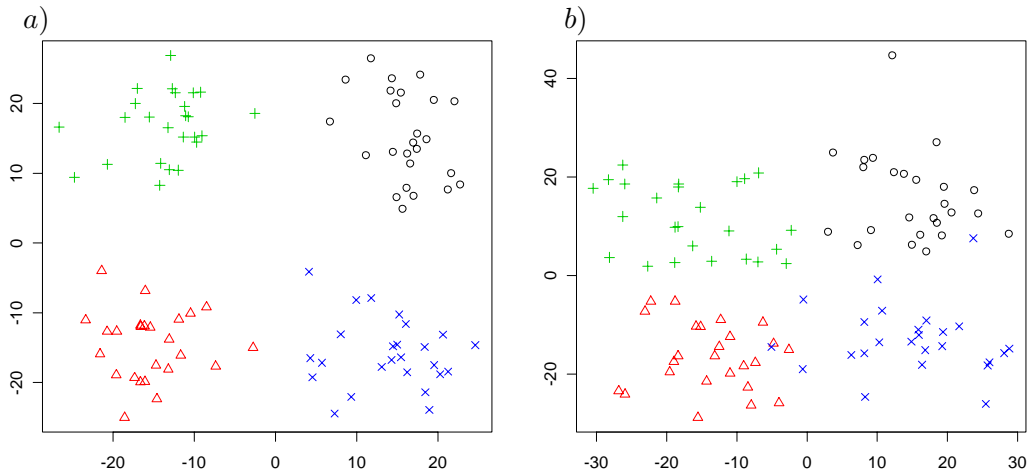
³W teorii pomiaru rozróżnia się 4 podstawowe skale pomiaru, wprowadzone przez Stevensa w 1959 roku, uporządkowane od najsłabszej do najmocniejszej: nominalna, porządkowa (rangowa), przedziałowa (interwałowa), ilorazowa (stosunkowa).

⁴BMI - współczynnik masy ciała (ang. Body Mass Index), obliczany jako iloraz masy ciała $[kg]$ i kwadratu wzrostu $[m^2]$. Klasyfikacja niedowagi, nadwagi oraz otyłości wyrażona w BMI dostępna jest na stronie <http://pl.wikipedia.org/wiki/BMI>.

⁵Kategoryzacja nadciśnienia tętniczego wyrażona w wartościach ciśnienia skurczowego i rozkurczowego dostępna jest na stronie <http://www.nadcisnienietetnicze.pl>.

podziału na 2 populacje/klasy: „nie” - pacjent dotąd nie zachorował na zawał serca, „tak” - pacjent przeżył zawał serca.

Podkreślmy, że dwie obserwacje $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{X}$ o identycznych wartościach atrybutów ($\mathbf{x}_1 = \mathbf{x}_2$) mogą należeć do różnych klas. Jest bardzo prawdopodobne, że istnieją dwie różne osoby tej samej płci, w identycznym wieku, z jednakowym BMI, obwodem pasa oraz typem nadciśnienia, i pierwsza z nich przeżyła zawał serca, natomiast druga jest zdrowa.



Rysunek 1.1: Przykładowe zbiory danych. Na każdym rysunku po 100 obserwacji z czterech różnych dwuwymiarowych rozkładów normalnych, a) klasy są rozłączne, b) klasy nie są rozłączne.

Przykład 1.2.2 Na rysunku 1.1 przedstawiono 100 obserwacji z czterech różnych dwuwymiarowych rozkładów normalnych. Współrzędne punktów płaszczyzny dostarczają informacji o dwóch ciągłych atrybutów.

Zadanie *klasyfikacji pod nadzorem* polega na wyborze *reguły decyzyjnej* (*reguły dyskryminacyjnej*, *klasyfikatora*),

$$d(\cdot, \mathcal{L}) : \mathbb{X} \rightarrow \{1, 2, \dots, g\} \quad \text{gdzie} \quad \mathbb{X} \ni \mathbf{x} \mapsto d(\mathbf{x}, \mathcal{L}) \in \{1, 2, \dots, g\} \quad (1.5)$$

przypisującej dowolnej obserwacji $\mathbf{x} \in \mathbb{X}$ przynależność do klasy o etykiecie $d(\mathbf{x}, \mathcal{L}) \in \{1, 2, \dots, g\}$.

Wyboru reguły dokonuje się na podstawie zbioru dostępnych obserwacji \mathcal{L} zwanych *próbą uczącą* i czasami *zbiorem uczącym*. Jeżeli \mathbb{L} jest zbiorem prób uczących to klasyfikator d przyjmuje postać odwzorowania produktu $\mathbb{X} \times \mathbb{L}$ w zbiór etykiet klas $\{1, 2, \dots, g\}$. Tam, gdzie nie będzie prowadziło to do nieporozumień, zamiast $d(\mathbf{x}, \mathcal{L})$ pisać będziemy krótko $d(\mathbf{x})$.

Definicja 1.2.1 Każdą funkcję h postaci

$$h : \mathbb{X} \rightarrow \{1, 2, \dots, g\} \quad \text{gdzie} \quad \mathbb{X} \ni \mathbf{x} \mapsto h(\mathbf{x}) \in \{1, 2, \dots, g\} \quad (1.6)$$

nazywamy *hipotezą*.

Hipoteza jest funkcją jedynie zmiennej \mathbf{x} , zatem reguła (1.5) jest hipotezą. Reguła (1.5) przedstawia „nauczony” klasyfikator. Zadanie klasyfikacji pod nadzorem nazywane jest również *klasyfikacją z próbą uczącą* i czasami *klasyfikacją z nauczycielem*. Podział obserwacji w

zbiorze \mathcal{L} na populację umożliwia (w pewnym stopniu) wykrycie zależności pomiędzy wartościami atrybutów a przynależnością do klasy. *Analiza dyskryminacyjna*⁶ dostarcza metod umożliwiających rozwiązanie tego zadania.

Próba ucząca umożliwia konstrukcję reguły. Głównym zadaniem klasyfikacji jest wykorzystanie reguły do predykcji. Każdej nowej obserwacji (z nieznaną przynależnością do populacji) chcemy przydzielić pewną (domniemaną) klasę, popełniając możliwie mały błąd. Posiadając dostatecznie dużą liczbę danych, w celu oszacowania błędu, próbę uczącą dzieli się na *podpróbę uczącą* oraz *podpróbę testową*. Reguła konstruowana jest jedynie na podstawie podpróby uczącej. Etykietowane obserwacje należące do podpróby testowej umożliwiają oszacowanie jakości predykcji (liczba błędnych klasyfikacji dokonanych na podpobie testowej, porównanie oryginalnej etykiety z etykietą nadaną przez klasyfikator). Szerzej o metodach oceny jakości klasyfikacji piszemy w sekcji 1.7.

Reguła dyskryminacyjna wprowadza podział zbioru obserwacji na g rozłącznych podzbiorów

$$\mathbb{X}_k := d^{-1}(k) = \{\mathbf{x} \in \mathbb{X} : d(\mathbf{x}) = k\} \quad (1.7)$$

gdzie brzegi każdego z podzbiorów mogą rozdzielać klasy.

Klasyfikacja pod nadzorem jest bardzo szeroko wykorzystywana w świecie nauki, przemysłu, biznesu. Podleganie ryzyku zachorowania na daną chorobę, wykrywanie nadużyć związanych z kradzieżą energii elektrycznej, zdolność kredytowa klientów banku, maszynowe rozpoznawanie pisma, niechciane wiadomości (SPAM) - każdy z tych problemów można przedstawić jako zadanie klasyfikacji pod nadzorem, jeżeli posiadamy dostateczną ilość danych wraz z precyzyjną definicją grup. Należy podkreślić, że w praktyce procesy klasyfikacyjne obciążone są niepewnością wynikającą naogół z braku rozdzielnosci klas. Dwie identyczne obserwacje w próbie uczącej pochodzące z różnych klas świadczą najczęściej o tym, że nośniki rozkładów wektorów losowych w różnych klasach nie są rozłączne.

Czasami klasyfikator d wygodnie jest przedstawiać jako wektor funkcji wskaźnikowych.

Definicja 1.2.2 Postacią wskaźnikową klasyfikatora $d : \mathbb{X} \rightarrow \{1, 2, \dots, g\}$ nazywamy wektor (c_1, c_2, \dots, c_g) funkcji wskaźnikowych, gdzie

$$c_k : \mathbb{X} \rightarrow \{0, 1\} \quad c_k(\mathbf{x}) := \begin{cases} 1 & \text{gdy } d(\mathbf{x}) = k \\ 0 & \text{gdy } d(\mathbf{x}) \neq k \end{cases} \quad \text{dla } k \in \{1, 2, \dots, g\} \quad (1.8)$$

Przy ocenie jakości klasyfikacji dokonywanej przez dany klasyfikator należy wprowadzić w zbiorze klasyfikatorów porządek liniowy pozwalający na porównanie dowolnych dwóch z nich.

Definicja 1.2.3 Funkcją straty (*stratą*) związaną z zaklasyfikowaniem obserwacji z klasy $i \in \{1, 2, \dots, g\}$ do klasy $j \in \{1, 2, \dots, g\}$ nazywamy funkcję

$$L : \{1, 2, \dots, g\}^2 \rightarrow \{0, 1\} \quad L(i, j) := \begin{cases} 0 & \text{gdy } i = j \\ 1 & \text{gdy } i \neq j \end{cases} \quad (1.9)$$

Przyjęliśmy zerowy koszt decyzji poprawnej oraz stały jednostkowy koszt decyzji błędnej. W praktyce często koszty decyzji błędnych nie są stałe.

⁶Klasyfikacja pod nadzorem jest szczególnym przypadkiem analizy dyskryminacyjnej. Oprócz zadania klasyfikacji (podziału) analiza dyskryminacyjna zajmuje się opisaniem różnic pomiędzy klasami (populacjami).

Definicja 1.2.4 Ryzykiem klasyfikatora $d : \mathbb{X} \rightarrow \{1, 2, \dots, g\}$, gdy oczekiwaną klasą jest klasa $k \in \{1, 2, \dots, g\}$ nazywamy oczekiwany koszt (oczekiwaną stratę) $R(d, k)$ zaklasyfikowania przez klasyfikator d obserwacji losowej z klasy k

$$R(d, k) := \mathbb{E}_{\mathbf{x}}[L(k, d(\mathbf{x})) \mid y = k] \quad (1.10)$$

Zauważmy, że

$$R(d, k) = \sum_{r=1}^g L(k, r) P[d(\mathbf{x}) = r \mid y = k] = P[d(\mathbf{x}) \neq k \mid y = k] \quad (1.11)$$

Ryzyko klasyfikatora, gdy prawdziwą jest klasa k , jest równe prawdopodobieństwu błędnej klasyfikacji losowej obserwacji do klasy innej niż k .

Definicja 1.2.5 Ryzyko całkowite klasyfikatora $d : \mathbb{X} \rightarrow \{1, 2, \dots, g\}$ określamy jako

$$R(d) := \mathbb{E}_k[R(d, k)] \quad (1.12)$$

Ryzyko całkowite otrzymujemy uznając klasę obserwacji za losową. Im mniejsze jest ryzyko całkowite tym mniejszy jest błąd klasyfikacji oraz tym lepszy jest klasyfikator. W tym sensie ryzyko całkowite R wprowadza porządek liniowy w zbiorze możliwych klasyfikatorów.

$$d_1 \preceq d_2 \Leftrightarrow R(d_1) \leq R(d_2)$$

W bardziej ogólnym przypadku, gdzie klasyfikator jest funkcją $d(\cdot, \mathcal{L}) : \mathbb{X} \rightarrow \{1, 2, \dots, g\}$, zbiór uczący $\mathcal{L} \in \mathbb{L}$ możemy uznać za losowy. Ryzyko całkowite takiego klasyfikatora definiujemy następująco

$$R_{\mathcal{L}}(d) := R(d(\cdot, \mathcal{L})) \quad (1.13)$$

Oczekiwane ryzyko całkowite

$$R^*(d) := \mathbb{E}_{\mathcal{L}}[R_{\mathcal{L}}(d)] \quad (1.14)$$

określa ciekawą miarę jakości takiego klasyfikatora.

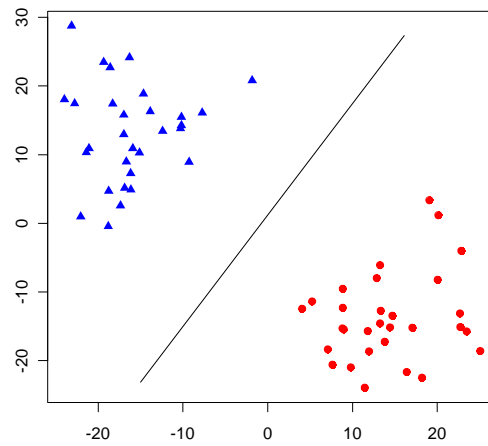
1.3. Liniowa funkcja dyskryminacyjna

Okazuje się, że dla przypadku $\mathbb{X} \subseteq \mathbb{R}^p$ często istnieją hiperpowierzchnie umożliwiające dobry podział \mathbb{X} na klasy. Rysunek 1.2 przedstawia obserwacje w \mathbb{R}^2 z dwóch klas rozdzielone prostą (hiperpowierzchnią w \mathbb{R}^2). Rysunek 1.3 przedstawia obserwacje w \mathbb{R}^3 z dwóch klas rozdzielone płaszczyzną (hiperpowierzchnią w \mathbb{R}^3). Metody generujące w zbiorze obserwacji \mathbb{X} liniowe hiperpowierzchnie dzielące klasy nazywamy *metodami liniowymi*. Praktyka pokazuje, że są to metody proste, często dające dobre rozwiązania (szczególnie dla obserwacji z wielowymiarowych rozkładów normalnych).

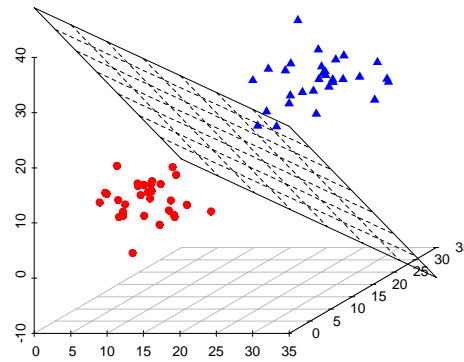
1.3.1. Problem dwóch klas

W 1936 roku sir Ronald Fisher⁷ zaproponował metodą liniową dla problemu dwóch klas ($g = 2$). Pomysł sir Fishera polegał na znalezieniu takiego kierunku $\mathbf{a} \in \mathbb{X} \subseteq \mathbb{R}^p$, że rzutowanie ortogonalne obserwacji na ten kierunek daje najlepszy możliwy rozdział na populacje. Algorytm ten nazywany jest algorytmem *liniowej analizy dyskryminacyjnej* (w skrócie *LDA* od

⁷Sir Ronald Aylmer Fisher - brytyjski naukowiec, zajmujący się statystyką matematyczną, biologią i genetyką. Biografia sir Fishera dostępna jest na stronie http://en.wikipedia.org/wiki/Ronald_Fisher.



Rysunek 1.2: Dwie rozłączne klasy rozdzielone prostą dyskryminacyjną (hiperpowierzchnią w \mathbb{R}^2).



Rysunek 1.3: Dwie rozłączne klasy rozdzielone płaszczyzną dyskryminacyjną (hiperpowierzchnią w \mathbb{R}^3).

anglojęzycznego terminu *linear discriminant analysis*). Poniżej przedstawimy jedynie główne kroki konstrukcji klasyfikatora LDA, szczegółowy opis metody znajduje się w [1] rozdział 1.

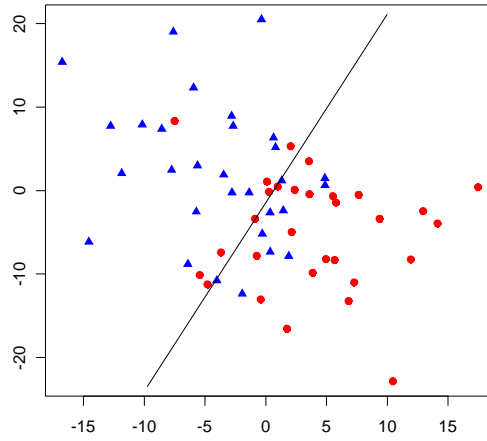
Sir Fisher tak sformułował zadanie analizy dyskryminacyjnej: znajdź kierunek $\mathbf{a} \in \mathbb{X}$, który najlepiej rozdziela klasy w próbie uczącej, przy tym, konstruując miarę odległości między klasami uwzględnij zmienność wewnątrzgrupową (wewnątrz klas).

Metoda LDA wymaga zagregowania informacji o klasach, do których będą klasyfikowane nowe obserwacje, przy użyciu wskaźników położenia i rozproszenia dla klas w próbie uczącej. Obserwacje są wektorami i należy rozpatrywać wektorową wartość oczekiwaną (wektorową średnią próbkową) oraz macierz kowariancji (próbkową macierz kowariancji).

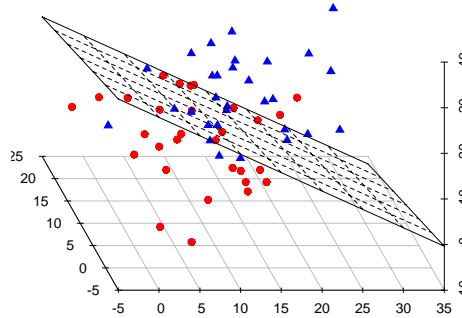
Rozpatrywaną próbę uczącą (\mathbf{x}_i, y_i) , $i = 1, 2, \dots, n$ dzielimy na dwie podpróby:

$$\begin{array}{ll} \mathbf{x}_{11}, \mathbf{x}_{12}, \dots, \mathbf{x}_{1n_1} & \text{z klasy 1} \\ \mathbf{x}_{21}, \mathbf{x}_{22}, \dots, \mathbf{x}_{2n_2} & \text{z klasy 2} \end{array} \quad (1.15)$$

gdzie $n_1 + n_2 = n$ (n_1 obserwacji z klasy 1, n_2 obserwacji z klasy 2). Średnie klas zapisujemy



Rysunek 1.4: Dwie nierozłączne klasy rozdzielone prostą dyskryminacyjną (hiperpowierzchnią w \mathbb{R}^2).



Rysunek 1.5: Dwie nierozłączne klasy rozdzielone płaszczyzną dyskryminacyjną (hiperpowierzchnią w \mathbb{R}^3).

jako:

$$\bar{\mathbf{x}}_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbf{x}_{1i} \quad \bar{\mathbf{x}}_2 = \frac{1}{n_2} \sum_{i=1}^{n_2} \mathbf{x}_{2i}$$

Próbkowe macierze kowariancji charakteryzują rozproszenie wewnątrzgrupowe (wewnątrz klas). Współczesna metoda LDA, jak i metoda zaproponowana przez Fishera opiera się na założeniu, że klasy posiadają taką samą macierz kowariancji. Macierz kowariancji wewnątrzgrupowej, wspólnej dla obydwu klas przyjmuje postać:

$$\mathbf{W} = \frac{1}{n-2} \sum_{k=1}^2 (n_k - 1) \mathbf{S}_k = \frac{1}{n-2} \sum_{k=1}^2 \left[\sum_{i=1}^{n_k} (\mathbf{x}_{ki} - \bar{\mathbf{x}}_k)(\mathbf{x}_{ki} - \bar{\mathbf{x}}_k)^T \right], \quad (1.16)$$

$$n = n_1 + n_2,$$

gdzie \mathbf{S}_k , $k = 1, 2$, są próbkowymi macierzami kowariancji w klasach 1 i 2. Próbkową miarą zmienności wewnątrzgrupowej wzdłuż kierunku \mathbf{a} jest wielkość

$$\mathbf{a}^T \mathbf{W} \mathbf{a} \quad (1.17)$$

Zgodnie z powyższym, zadaniu znalezienia reguły dyskryminacyjnej, sformułowanemu przez Fishera, nadajemy postać:

- Znajdź kierunek $\tilde{\mathbf{a}}$ w \mathbb{X} , który najlepiej rozdziela obydwie podpróby uczące, za miarę rozdzielności klas wzdłuż danego kierunku \mathbf{a} weź kwadrat odległości między średnimi w podpróbach wzdłuż tego kierunku,

$$(\mathbf{a}^T \bar{\mathbf{x}}_2 - \mathbf{a}^T \bar{\mathbf{x}}_1)^2$$

zmodyfikowany przez odpowiednio uwzględnioną zmienność wewnątrz klas wzdłuż kierunku \mathbf{a}

$$\frac{(\mathbf{a}^T \bar{\mathbf{x}}_2 - \mathbf{a}^T \bar{\mathbf{x}}_1)^2}{\mathbf{a}^T \mathbf{W} \mathbf{a}} \quad (1.18)$$

gdzie \mathbf{W} jest próbkową macierzą kowariancji wewnątrzgrupowej. Znalezienie kierunku najlepiej rozdzielającego klasy jest równoważne znalezieniu wektora kierunkowego $\tilde{\mathbf{a}}$ maksymalizującego wyrażenie 1.18.

- Mając kierunek $\tilde{\mathbf{a}}$ rzutuj ortogonalnie obydwie średnie klas oraz obserwację \mathbf{x} o nieznaną klasę na ten kierunek, zaklasyfikuj \mathbf{x} do klasy j jeżeli

$$|\tilde{\mathbf{a}}^T \mathbf{x} - \tilde{\mathbf{a}}^T \mathbf{x}_j| < |\tilde{\mathbf{a}}^T \mathbf{x} - \tilde{\mathbf{a}}^T \mathbf{x}_k| \quad (1.19)$$

dla $k \neq j, j, k \in \{1, 2\}$.

Fisher w swoim rozwiązaniu założył, że macierze kowariancji wewnątrz klas są identyczne. Okazuje się, że tą metodą można otrzymać dobre wyniki nawet wtedy, gdy założenie to nie jest spełnione.

Można wykazać, że wektor maksymalizujący iloraz 1.18 jest proporcjonalny do wektora $\mathbf{W}^{-1}(\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1)$:

$$\tilde{\mathbf{a}} \propto \mathbf{W}^{-1}(\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1) \quad (1.20)$$

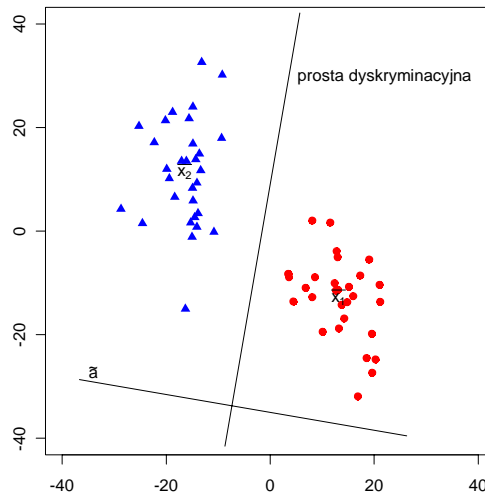
Zmienna $\tilde{\mathbf{a}}^T \mathbf{x}$ nazywana jest (*pierwszą*) *zmienną kanoniczną*, natomiast wektor $\tilde{\mathbf{a}}$ (*pierwszym*) *wektorem kanonicznym*.

W niniejszej pracy nie będziemy zajmować się problemem wielu klas. Szczegółowy opis uogólnienia metody LDA na wiele klas można znaleźć w [1].

1.4. Dyskryminacja oparta na rozkładach prawdopodobieństwa

Konstruując metodę LDA przedstawiliśmy podejście geometryczne zakładając, że klasy rozdzielimy liniowymi hiperpowierzchniami. We wstępie wspomnieliśmy również, że w praktyce obserwowane klasy często nie są rozdzielne. Klasy, gdzie nośniki rozkładów na siebie „zachodzą”, odróżnia jedynie postać rozkładów. W takim przypadku naturalnym staje się podejście probabilistyczne oparte na rozkładach prawdopodobieństw obserwacji w klasach.

Rozkłady w populacjach dzielimy na trzy typy. W pierwszym posiadamy pełną informację o postaci rozkładu i jego parametrach. Drugi jest modyfikacją pierwszego, tzn. znana jest postać rozkładu, lecz parametry rozkładu muszą być estymowane. Trzeci typ jest czystym



Rysunek 1.6: Ortogonalny rzut obserwacji na prostą wyznaczoną kierunkiem \tilde{a} oraz przecięcie tej prostej z prostą dyskryminacyjną dobrze rozdziela klasy.

podejściem empirycznym, gdzie nie zakładamy nic o rozkładach w klasach. W praktyce spotykamy najczęściej typ drugi i trzeci.

W poniższym tekście przyjmujemy, że rozkład obserwacji $\mathbf{x} \in \mathbb{X}$ z klasy $k \in \{1, 2, \dots, g\}$ jest dany dyskretnym rozkładem prawdopodobieństwa $p(\mathbf{x}|k)$ lub funkcją gęstości prawdopodobieństwa $p(\mathbf{x}|k) \equiv f_k(x)$. Przypadek rozkładów dyskretno-ciągłych pomijamy.

1.4.1. Reguła największej wiarygodności

Pierwszą i najbardziej intuicyjną regułą dyskryminacyjną opartą na rozkładach prawdopodobieństw obserwacji w klasach jest metoda *największej wiarygodności*.

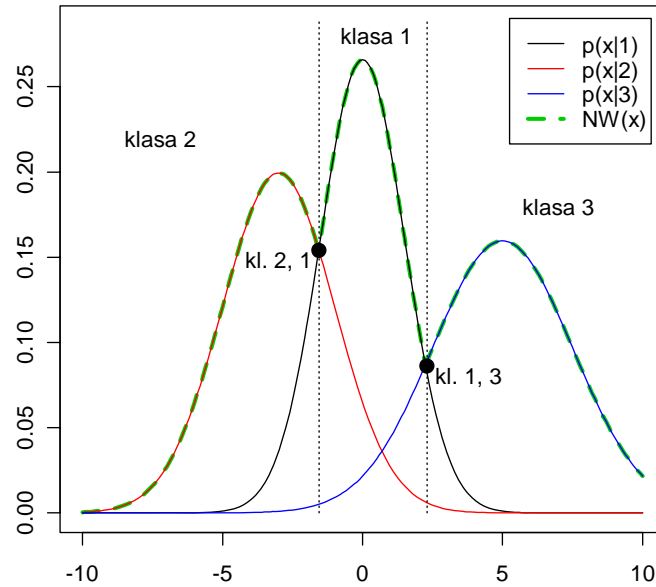
Definicja 1.4.1 Regułę dyskryminacyjną $d : \mathbb{X} \rightarrow \{1, 2, \dots, g\}$ daną wzorem

$$d(\mathbf{x}) \equiv \text{nw}(\mathbf{x}) = \arg \max_k p(\mathbf{x}|k) \quad (1.21)$$

nazywamy regułą *największej wiarygodności* (w skrócie NW).

W przypadku gęstości prawdopodobieństwa reguła NW dla obserwacji $\mathbf{x} \in \mathbb{X}$ wybiera tę klasę $k \in \{1, 2, \dots, g\}$, dla której wartość gęstości $p(\mathbf{x}|k) \equiv f_k(\mathbf{x})$ jest najwyższa. Przypadek dyskretny nie wymaga wyjaśnień. Problem pojawia się, gdy nie można jednoznacznie określić, do której z klas przypisać \mathbf{x} , tzn. kiedy istnieją $i, j \in \{1, 2, \dots, g\}$, że $i \neq j$ oraz $p(\mathbf{x}|i) = p(\mathbf{x}|j) = \max_k p(\mathbf{x}|k)$. W takim przypadku obserwację \mathbf{x} przyporządkowuje się w sposób losowy do jednej z klas $\{i : p(\mathbf{x}|i) = \max_k p(\mathbf{x}|k)\}$.

Rysunek 1.7 przedstawia trzy gęstości: $p(\mathbf{x}|1)$, $p(\mathbf{x}|2)$, $p(\mathbf{x}|3)$. Krzywa $NW(\mathbf{x})$ wskazuje klasę, do której reguła NW przydziela obserwację \mathbf{x} . Za wyjątkiem dwóch punktów reguła NW jednoznacznie klasyfikuje obserwacje $\mathbf{x} \in \mathbb{X}$ do klas 1, 2 lub 3. Na wykresie zaznaczono również dwie obserwacje $\mathbf{x}_1, \mathbf{x}_2$, dla których nie można jednoznacznie określić przynależności do klasy. Losowy wybór klasy spośród klas 2 lub 1 konieczny jest dla obserwacji \mathbf{x}_1 , tzn. $p(\mathbf{x}_1|2) = p(\mathbf{x}_1|1) = NW(\mathbf{x}_1)$. Losowy wybór klasy spośród klas 1 lub 3 konieczny jest dla obserwacji \mathbf{x}_2 , tzn. $p(\mathbf{x}_2|1) = p(\mathbf{x}_2|3) = NW(\mathbf{x}_2)$.



Rysunek 1.7: Trzy gęstości: $p(\mathbf{x}|1)$, $p(\mathbf{x}|2)$, $p(\mathbf{x}|3)$. Krzywa $NW(\mathbf{x})$ wskazuje klasę, do której reguła NW przydziela obserwację \mathbf{x} .

1.4.2. Reguła Bayesa

Do założeń odnośnie rozkładu obserwacji $\mathbf{x} \in \mathbb{X}$ z klasy $k \in \{1, 2, \dots, g\}$ dodajmy prawdopodobieństwa *a priori*⁸ $\pi_1, \pi_2, \dots, \pi_g$, że obserwacja (dowolna) pochodzi z klasy k . Prawdopodobieństwa *a priori* mogą być oszacowane w przypadku posiadania dostatecznie dużej próby uczącej przez wyznaczenie częstości (proporcji) w próbie uczącej obserwacji z danej klasy.

Na mocy twierdzenia Bayesa prawdopodobieństwo *a posteriori*⁹, że zaobserwowana wartość $\mathbf{x} \in \mathbb{X}$ pochodzi klasy $k \in \{1, 2, \dots, g\}$ wynosi

$$p(k|\mathbf{x}) = \frac{\pi_k p(\mathbf{x}|k)}{\sum_{r=1}^g \pi_r p(\mathbf{x}|r)} \quad (1.22)$$

Przy znanym prawdopodobieństwie *a posteriori* $p(k|\mathbf{x})$ najbardziej naturalną regułą dyskryminacyjną jest *reguła Bayesa*.

Definicja 1.4.2 Regułę dyskryminacyjną $d: \mathbb{X} \rightarrow \{1, 2, \dots, g\}$ daną wzorem

$$d(\mathbf{x}) \equiv \mathbf{b}(\mathbf{x}) = \arg \max_k p(k|\mathbf{x}) \quad (1.23)$$

nazywamy regułą Bayesa (klasyfikatorem Bayesa).

Reguła Bayesa dla obserwacji $\mathbf{x} \in \mathbb{X}$ wybiera tę klasę $k \in \{1, 2, \dots, g\}$, dla której największe jest prawdopodobieństwo $p(k|\mathbf{x})$. Podobnie jak w przypadku reguły NW pojawia się problem jednoznaczności przydzielenia \mathbf{x} do klasy w sytuacji, gdy istnieją $i, j \in \{1, 2, \dots, g\}$, że $i \neq j$ oraz $p(i|\mathbf{x}) = p(j|\mathbf{x}) = \max_k p(k|\mathbf{x})$. Takie obserwacje przyporządkowuje się w sposób losowy do jednej z klas $\{i : p(i|\mathbf{x}) = \max_k p(k|\mathbf{x})\}$.

⁸A priori - z założenia.

⁹A posteriori - z następstwa.

Stwierdzenie 1.4.1 Jeżeli znane są prawdopodobieństwa a priori $\pi_1, \pi_2, \dots, \pi_g$ oraz prawdopodobieństwa (funkcje gęstości) $p(\mathbf{x}|k)$, to na mocy (1.22) wybór takiej klasy $k \in \{1, 2, \dots, g\}$, że maksymalne jest prawdopodobieństwo $p(k|\mathbf{x})$, równoważny jest wyborowi takiego k , że maksymalna jest wartość wyrażenia $\pi_k p(\mathbf{x}|k)$.

Wniosek 1.4.1 Zachodzi równość

$$\mathbf{b}(\mathbf{x}) = \arg \max_k \pi_k p(\mathbf{x}|k) \quad (1.24)$$

Wniosek 1.4.2 Reguła NW jest regułą Bayesa w przypadku równych prawdopodobieństw a priori $\pi_1 = \pi_2 = \dots = \pi_g$.

Okazuje się, że żaden klasyfikator nie może mieć mniejszego ryzyka całkowitego niż ryzyko całkowite klasyfikatora Bayesowskiego.

Twierdzenie 1.4.1 Reguła Bayesa minimalizuje ryzyko całkowite.

Dowód twierdzenia przeprowadzimy jedynie dla przypadku ciągłego rozkładu obserwacji losowych \mathbf{x} , gdy $\mathbb{X} = \mathbb{R}^p$.

Dowód: Niech będzie dana postać wskaźnikowa (c_1, c_2, \dots, c_g) (def. 1.2.2) dowolnego ustalonego klasyfikatora $d : \mathbb{R}^p \rightarrow \{1, 2, \dots, g\}$. Oznaczmy przez p_{ij} prawdopodobieństwo zaklasyfikowania przez klasyfikator d obserwacji losowej z klasy $i \in \{1, 2, \dots, g\}$ do klasy $j \in \{1, 2, \dots, g\}$. Rozkład losowej obserwacji \mathbf{x} z klasy k dany jest funkcją gęstości prawdopodobieństwa $p(\mathbf{x}|k)$. Zatem możemy zapisać

$$p_{ij} = \int_{\mathbb{R}^p} \mathbb{I}_{[d(\mathbf{x})=j]} p(\mathbf{x}|i) d\mathbf{x} = \int_{\mathbb{R}^p} c_j(\mathbf{x}) p(\mathbf{x}|i) d\mathbf{x}$$

Ryzyko całkowite (def. 1.2.5) klasyfikatora d na podstawie (1.11) przyjmuje postać

$$R(d) = \mathbb{E}_k R(d, k) = \sum_{k=1}^g \pi_k P[d(\mathbf{x}) \neq k \mid y = k] = \sum_{k=1}^g \pi_k (1 - p_{kk}) = 1 - \sum_{k=1}^g \pi_k p_{kk}$$

Rozpatrzmy dalej postać wskaźnikową $(c_1^*, c_2^*, \dots, c_g^*)$ klasyfikatora Bayesowskiego $\mathbf{b} : \mathbb{R}^p \rightarrow \{1, 2, \dots, g\}$. Oczywiście

$$c_i^*(\mathbf{x}) := \begin{cases} 1 & \text{gdy } \pi_i p(\mathbf{x}|i) = \max_k \pi_k p(\mathbf{x}|k) \\ 0 & \text{w przypadku przeciwnym.} \end{cases} \quad \text{dla } i \in \{1, 2, \dots, g\}$$

Zauważmy, że

$$\begin{aligned} \sum_{k=1}^g \pi_k p_{kk} &= \sum_{k=1}^g \int_{\mathbb{R}^p} c_k(\mathbf{x}) \pi_k p(\mathbf{x}|k) d\mathbf{x} \leq \sum_{k=1}^g \int_{\mathbb{R}^p} c_k(\mathbf{x}) \max_j \pi_j p(\mathbf{x}|j) d\mathbf{x} = \\ &= \int_{\mathbb{R}^p} \left(\sum_{k=1}^g c_k(\mathbf{x}) \right) \max_j \pi_j p(\mathbf{x}|j) d\mathbf{x} = \int_{\mathbb{R}^p} \max_j \pi_j p(\mathbf{x}|j) d\mathbf{x} = \sum_{k=1}^g \int_{\mathbb{R}^p} c_k^*(\mathbf{x}) \pi_k p(\mathbf{x}|k) d\mathbf{x} = \\ &= \sum_{k=1}^g p_{kk}^* \end{aligned}$$

ponieważ

$$\max_j \pi_j p(\mathbf{x}|j) = \sum_{k=1}^g c_k^*(\mathbf{x}) \pi_k p(\mathbf{x}|k)$$

oraz przyjmując, że

$$p_{ij}^* = \int_{\mathbb{R}^p} \mathbb{I}_{[\mathbf{b}(\mathbf{x})=j]} p(\mathbf{x}|i) d\mathbf{x} = \int_{\mathbb{R}^p} c_j^*(\mathbf{x}) p(\mathbf{x}|i) d\mathbf{x}$$

jest prawdopodobieństwo zaklasyfikowania przez klasyfikator bayesowski \mathbf{b} obserwacji losowej z klasy $i \in \{1, 2, \dots, g\}$ do klasy $j \in \{1, 2, \dots, g\}$. Zatem

$$R(d) = 1 - \sum_{k=1}^g \pi_k p_{kk} \geq 1 - \sum_{k=1}^g \pi_k p_{kk}^* = R(\mathbf{b})$$

■

Ryzyko klasyfikatora Bayesa nazywamy *ryzykiem bayesowskim*. Podkreślmy, że dla $\mathbb{X} = \mathbb{R}^p$ oraz ciągłego rozkładu $p(\mathbf{x}|k)$ obserwacji losowych \mathbf{x} z klasy k ryzyko bayesowskie dane jest wzorem

$$R(\mathbf{b}) = 1 - \sum_{k=1}^g \int_{\mathbb{R}^p} \mathbb{I}_{[\mathbf{b}(\mathbf{x})=k]} \pi_k p(\mathbf{x}|k) d\mathbf{x} = 1 - \int_{\mathbb{R}^p} \max_k \pi_k p(\mathbf{x}|k) d\mathbf{x} \quad (1.25)$$

1.5. Dyskryminacja jako problem regresji

W sekcji tej nie będziemy zajmować się szczegółowym przedstawieniem metod rozwiązania zadania dyskryminacji przez zadanie regresji. Zwrócimy jedynie uwagę na główny związek pomiędzy zagadnieniami.

Zadanie dyskryminacji jest w istocie zadaniem estymacji funkcji o wartościach nominalnych. W przypadku, gdy liczba klas $g = 2$, a zbiór klas to $\{0, 1\}$, z łatwością zauważymy, że

$$\mathbb{E}(y|\mathbf{x}) = P(y = 1|\mathbf{x}) \quad (1.26)$$

gdzie $\mathbf{x} \in \mathbb{X}$ jest losową obserwacją, $\mathbb{E}(y|\mathbf{x})$ jest warunkową wartością oczekiwaną zmiennej losowej y pod warunkiem zaobserwowania \mathbf{x} , a $P(y = 1|\mathbf{x})$ jest prawdopodobieństwem warunkowym, że znajdujemy się w klasie 1 pod warunkiem zaobserwowania \mathbf{x} .

Przypomnijmy, że dla zmiennej niezależnej (zmiennej objaśniającej) X oraz zmiennej zależnej (zmiennej objaśnianej) Y , funkcja regresji, a dokładniej *linia regresji pierwszego rodzaju*, dana jest wartością oczekiwaną

$$\mathbb{E}(Y|X = x)$$

Zadanie dyskryminacji można zatem przedstawić jako zadanie regresji z funkcją regresji $P(y = 1|\mathbf{x}) = p(1|\mathbf{x})$. Dla problemu z liczbą klas $g > 2$ istnieje związek z regresją wielowymiarową. Stosując kodowanie etykiety klasy dla obserwacji \mathbf{x} z klasy $k \in \{1, 2, \dots, g\}$ przez wektor wskaźnikowy $\mathbf{y} = (y^{(1)}, y^{(2)}, \dots, y^{(g)})$

$$y^{(i)} := \begin{cases} 1 & \text{gdy } i = k \\ 0 & \text{gdy } i \neq k \end{cases} \quad \text{dla } i \in \{1, 2, \dots, g\}$$

bezpośrednio otrzymujemy

$$\mathbb{E}(y^{(k)}|\mathbf{x}) = p(k|\mathbf{x}) \quad (1.27)$$

dla $k \in \{1, 2, \dots, g\}$.

1.6. Dyskryminacja logistyczna

Rozważmy zadanie dyskryminacji dla problemu 2-klasowego (klasy kodowane ich numerami). W podejściu *logistycznym* estymujemy prawdopodobieństwa $p(k|\mathbf{x})$, że zaobserwowana wartość \mathbf{x} pochodzi z klasy $k \in \{1, 2\}$.

Model przyjmuje postać:

$$\ln \frac{\hat{p}(2|\mathbf{x})}{1 - \hat{p}(2|\mathbf{x})} = \alpha + \beta^T \mathbf{x}. \quad (1.28)$$

Rozviklując powyższy model otrzymujemy:

$$\hat{p}(2|\mathbf{x}) = \frac{e^{\alpha + \beta^T \mathbf{x}}}{1 + e^{\alpha + \beta^T \mathbf{x}}} \quad (1.29)$$

oraz

$$\hat{p}(1|\mathbf{x}) = \frac{1}{1 + e^{\alpha + \beta^T \mathbf{x}}}. \quad (1.30)$$

Funkcję $\ln \frac{v}{1-v}$ oznacza się często $\text{logit}(v)$ i nazywa *funkcją logitową*. Parametry (α, β) estymatorów (modelu) (1.29) oraz (1.30) wyznaczamy metodą największej wiarygodności, tzn. maksymalizując funkcję wiarygodności (1.31) na podstawie próby.

$$\prod_{i=1}^n \hat{p}(2|\mathbf{x}_i)^{y_i} \hat{p}(1|\mathbf{x}_i)^{1-y_i} \quad (1.31)$$

W powyższym y_i jest wartością funkcji wskaźnikowej klasy 2 dla i -tej obserwacji.

W momencie, gdy wyznaczone są estymatory $\hat{p}(1|\mathbf{x})$ oraz $\hat{p}(2|\mathbf{x})$, reguła dyskryminacyjna polega na wybraniu większej z tych wartości i odpowiedniej klasyfikacji obserwacji \mathbf{x} (reguła Bayesa).

Uogólniając powyższy model na przypadek wielu nadal zakładamy, że klasy kodujemy ich numerami. Konstrukcja estymatorów prawdopodobieństw $p(k|\mathbf{x})$ przebiega w poniższy sposób:

$$\begin{aligned} \ln \frac{\hat{p}(1|\mathbf{x})}{\hat{p}(g|\mathbf{x})} &= \beta_{10} + \beta_1^T \mathbf{x} \\ \ln \frac{\hat{p}(2|\mathbf{x})}{\hat{p}(g|\mathbf{x})} &= \beta_{20} + \beta_2^T \mathbf{x} \\ &\vdots \\ \ln \frac{\hat{p}(g-1|\mathbf{x})}{\hat{p}(g|\mathbf{x})} &= \beta_{(g-1)0} + \beta_{g-1}^T \mathbf{x} \end{aligned}$$

Ostatecznie:

$$\hat{p}(k|\mathbf{x}) = \frac{e^{(\beta_{k0} + \beta_k^T \mathbf{x})}}{1 + \sum_{l=1}^{g-1} e^{(\beta_{l0} + \beta_l^T \mathbf{x})}} \quad (1.32)$$

$k = 1, \dots, g-1$, oraz

$$\hat{p}(g|\mathbf{x}) = \frac{1}{1 + \sum_{l=1}^{g-1} e^{(\beta_{l0} + \beta_l^T \mathbf{x})}}. \quad (1.33)$$

Zauważmy, że

$$\sum_{l=1}^g \hat{p}(l|\mathbf{x}) = 1$$

Oznaczmy

$$\hat{p}(k|\mathbf{x}) = \hat{p}_k(\mathbf{x}, \boldsymbol{\theta})$$

gdzie $\boldsymbol{\theta} = \{\beta_{10}, \beta_1, \dots, \beta_{(g-1)0}, \beta_{g-1}\}$. Logarytm funkcji wiarygodności przyjmuje postać:

$$\sum_{i=1}^n \ln \hat{p}_{k_i}(\mathbf{x}_i, \boldsymbol{\theta}),$$

gdzie k_i jest klasą i -tego elementu próby. Następnie maksymalizując tę funkcję względem wektora $\boldsymbol{\theta}$ otrzymujemy wszystkie prawdopodobieństwa a posteriori. Reguła dyskryminacyjna w oczywisty sposób ma postać reguły bayesowskiej.

1.7. Ocena jakości klasyfikacji

Jakość procesu klasyfikacji rozpatrywana jest w kontekście własności zaproponowanego klasyfikatora. W typowym problemie dokonuje się oceny:

- błędu klasyfikacji, tzn. prawdopodobieństwa niepoprawnej decyzji,
- własności klasyfikatora podczas *procesu uczenia*,
- własności klasyfikatora podczas *procesu predykcji*.

W zależności od typu rozwiązywanego problemu powyższe punkty traktuje się z różną istotnością. Posiadając niewielką próbę uczącą możemy skupić się na minimalizacji błędu, zając się o czasie uczenia oraz predykcji. W praktyce efektywna analiza dużych zbiorów danych nie ma jednak szans powodzenia bez uwzględnienia wpływu wymienionych czasów. Jeszcze inna sytuacja zachodzi, gdy mamy do czynienia z różną dynamiką przyrostu danych uczących, gdzie własność skalowalności jest szczególnie ważna.

1.7.1. Błąd klasyfikacji

Oszacowanie błędu klasyfikacji pozwala wybrać klasyfikator możliwie najlepiej przybliżający dokładne rozwiązanie zadania dyskryminacji. Istnieje wiele technik oszacowania błędu klasyfikacji algorytmów uczących.

Ponowne podstawienie

Do budowy klasyfikatora i oceny jego skuteczności używany jest ten sam zbiór danych. Reguła klasyfikacyjna powstała przy użyciu zbioru uczącego testowana jest przez dokonanie predykcji na każdej obserwacji z próby uczącej¹⁰. Porównanie wyniku predykcji z oryginalną przynależnością do klasy pozwala wyznaczyć liczbę (procent) obserwacji błędnie zaklasyfikowanych. Jest to proste podejście, jednak taki estymator błędu może być mocno obciążony. Im bardziej złożony jest klasyfikator, tym mniejszy błąd dla próby uczącej jest generowany. Zbyt dokładne dopasowanie do danych uczących prowadzi do uwzględnienia przypadkowych zależności, a w konsekwencji do dużego błędu rzeczywistego. Metoda ponownego podstawienia daje zadawalające efekty jedynie dla prostych reguł, takich jak metody liniowe.

Próba testowa

W sytuacji, gdy dysponujemy odpowiednio licznym zbiorem uczącym dokonujemy losowania bez zwracania pewnej liczby jego elementów, tworząc próbę testową. Elementy pozostałe (nie ujęte w losowaniu) tworzą nową próbę uczącą, przy użyciu której konstruowany jest klasyfikator. Obserwacje z próby testowej, o których wiemy do jakich klas należą, poddawane są predykcji. Wynik predykcji w porównaniu z oryginalną przynależnością do klas daje oszacowanie prawdopodobieństwa niepoprawnej decyzji. Estymator uzyskany metodą próby testowej posiada na ogół mniejsze obciążenie. Wykorzystanie w celach testowych elementów nie biorących udziału w procesie uczenia dobrze odzwierciedla przypadek predykcji na danych rzeczywistych. Metoda próby testowej daje dobre rezultaty dla złożonych klasyfikatorów, takich jak drzewa klasyfikacyjne.

Krosvalidacja

Metodę krosvalidacji stosuje się dla małych zbiorów uczących¹¹. W k -krotnej krosvalidacji próbę uczącą dzieli się na k możliwie równolicznych części. W kolejnym kroku tworzone są *pseudopróby*, przy czym każda z nich jest sumą dowolnie różnych $k - 1$ części oryginalnej próby uczącej. Pseudopróby wykorzystywane są w procesie uczenia, czego wynikiem jest zbiór k klasyfikatorów (każdy z klasyfikatorów powstaje przy wykorzystaniu tylko jednej z pseudoprób, $\binom{k}{k-1} = k$ możliwości). Jakość każdego klasyfikatora oceniana jest przez sprawdzenie liczby błędnych klasyfikacji na tej części oryginalnego zbioru uczącego, która nie brała udziału w procesie uczenia danego klasyfikatora. Ostateczną jakość klasyfikacji podaje się jako średnią z wartości otrzymanych dla każdego klasyfikatora z osobna (wariancja wyników określa odchylenie od tej średniej). Estymator uzyskany metodą krosvalidacji jest „prawie” nieobciążony. Zazwyczaj mała liczba k prowadzi do większego obciążenia tego estymatora, ale zmniejsza jego wariancję. Szczególnym przypadkiem tej metody jest n -krotna krosvalidacja (z ang. *leave-one-out*), gdzie $k = n$ (n - liczność oryginalnej próby uczącej). Wyniki tej metody dają z reguły małe obciążenie estymatorów, prowadząc do dużej wariancji. Dla *stabilnych* klasyfikatorów (niewielkie zmiany w zbiorach uczących nie powodują znaczących zmiany w powstałych regułach decyzyjnych) metoda pozwala na otrzymanie zadawalających rezultatów. Oczywiście krosvalidacja jest kosztowna, dlatego też stosowana jest dla małych zbiorów danych.

¹⁰Dokonanie predykcji na każdej obserwacji z próby uczącej wiąże się z koniecznością ponownego wykorzystania danych uczących, stąd nazwa „ponowne podstawienie” (ang. *resubstitution*).

¹¹Breiman, Friedman, Olshen i Stone zaproponowali w [3], żeby używać krosvalidacji dopiero wtedy, gdy liczba elementów pochodzących z choć jednej klasy będzie mniejsza od 900.

Bootstrap

Niektóre modele klasyfikacyjne są *niestabilne*, co oznacza, że niewielkie zmiany w zbiorach uczących mogą powodować duże zmiany w powstałych regułach decyzyjnych. Dokładność niestabilnych modeli może zostać poprawiona metodą *bootstrap*. Technika ta polega na n -krotnym losowaniu ze zwracaniem pseudoprób, każda o liczebności n . Pseudopróby wykorzystywane są w procesie uczenia. W efekcie otrzymujemy kolejne wersje tego samego klasyfikatora. W następnym kroku dla każdego elementu z oryginalnej próby uczącej wyliczany jest procent błędnych klasyfikacji dokonanych przez te klasyfikatory, w budowie których dany element nie brał udziału. Następnie dla wszystkich elementów oryginalnej próby uczącej wyliczana jest średnia wartość otrzymanych proporcji błędnych klasyfikacji. Uzyskany wynik jest oszacowaniem prawdopodobieństwa niepoprawnej klasyfikacji. Otrzymane w ten sposób oszacowanie jest jednak obciążone. Pseudopróba typu bootstrap to model z losowaniem bez zwracania, zatem prawdopodobieństwo, że dany element nie zostanie do niej wylosowany wynosi

$$1 - \alpha = \underbrace{\left(1 - \frac{1}{n}\right) \left(1 - \frac{1}{n}\right) \dots \left(1 - \frac{1}{n}\right)}_{n\text{-losowań}} = \left(1 - \frac{1}{n}\right)^n \approx \frac{1}{e} = 0,368 \quad (1.34)$$

natomiast prawdopodobieństwo jego wylosowania dane jest liczbą

$$\alpha \approx 1 - \frac{1}{e} = 0,632 \quad (1.35)$$

Z powyższego bezpośrednio wynika, że podczas budowy kolejnych klasyfikatorów średnio wykorzystywane jest około $\frac{2}{3}$ obserwacji z oryginalnej próby uczącej, a średnio $\frac{1}{3}$ obserwacji z tej próby nie zostaje wylosowana do odpowiedniej pseudopróby, co w rezultacie powoduje zawyżenie otrzymanego oszacowania prawdopodobieństwa błędnej klasyfikacji (obciążenie dodatnie). Oznaczmy ten estymator symbolem $\hat{\theta}_{B+}$. Z drugiej strony, jeżeli rozważymy oszacowanie powyższego prawdopodobieństwa jako uśrednienie ułamków błędnych klasyfikacji otrzymanych dla wszystkich bootstrapowych wersji danego klasyfikatora, gdy każdy klasyfikator testowany jest na całej próbie uczącej, otrzymamy estymator $\hat{\theta}_{B-}$ o obciążeniu ujemnym (średnio aż około $\frac{2}{3}$ testowanych elementów brało udział w procesie uczenia). Rozpatrzmy estymator $\hat{\theta}_B$ dany kombinacją wypukłą

$$\hat{\theta}_B := \alpha \hat{\theta}_{B+} + (1 - \alpha) \hat{\theta}_{B-} \approx 0,632 \hat{\theta}_{B+} + 0,368 \hat{\theta}_{B-} \quad (1.36)$$

Okazuje się, że estymator $\hat{\theta}_B$, nazywany estymatorem *bootstrap 0,632*, koryguje obciążenia estymatorów $\hat{\theta}_{B+}$ i $\hat{\theta}_{B-}$ oraz, że jest godny polecenia w metodzie bootstrap. Estymator ten zawodzi w przypadkach, gdy zbiór uczący sprzyja nadmiernemu dopasowaniu się do danych uczących.

Uwagi praktyczne

W praktyce, podczas szacowania jakości klasyfikacji, bardzo przydatna okazuje się struktura zwana *macierzą błędów* (z ang. *confusion matrix*).

Definicja 1.7.1 Macierzą błędów dla zbioru etykietowanych obserwacji T , w problemie dyskryminacji o g klasach (klasy o etykietach $1, 2, \dots, g$), przy znanym klasyfikatorze d , nazywa

się $g \times g$ -macierz postaci

$$\begin{bmatrix} n_{11} & n_{12} & \dots & n_{1g} \\ n_{21} & n_{22} & \dots & n_{2g} \\ \vdots & \vdots & \ddots & \vdots \\ n_{g1} & n_{g2} & \dots & n_{gg} \end{bmatrix} \quad (1.37)$$

gdzie n_{ij} ($i, j = 1, \dots, g$) oznacza liczbę obserwacji w zbiorze T pochodzących z klasy i zaklasyfikowanych regułą d do klasy j .

Oczywiście $n = \sum_{i,j=1}^g n_{ij}$ jest liczbą elementów w zbiorze T . Z łatwością podajemy liczbę poprawnych klasyfikacji $\sum_{i=1}^g n_{ii}$ oraz liczbę klasyfikacji błędnych $\sum_{i,j=1, i \neq j}^g n_{ij}$. Podane wskaźniki naturalnie przenoszą się na analogiczne definicje w ograniczeniu do ustalonej klasy, co czasami ułatwia analizę jakości rozwiązania.

1.7.2. Proces uczenia oraz predykcji

Analiza procesu uczenia oraz predykcji pozwala na wybór klasyfikatora, który możliwie najlepiej odpowiada postawionym warunkom praktycznym zadania dyskryminacji (rozmiary danych, oczekiwane czasy uczenia oraz predykcji, zakładana dynamika przyrostu danych uczących).

Selekcja cech

Często informacje o obiekcie zbierane są w nadmiarze. W związku z tym występuje wiele zbędnych cech. Cechy mogą być zbędne, ponieważ nie wprowadzają żadnej nowej informacji lub też nie mają żadnego związku z celem klasyfikacji (przykładem są cechy liniowo zależne). Występowanie cech zbędnych powoduje niepotrzebne wydłużenie czasu uczenia oraz, w przypadku wielu modeli klasyfikacyjnych, prowadzi do trudności w poprawie jakości klasyfikacji. Dlatego też w realnych zastosowaniach klasyfikacyjnych przed przystąpieniem do uczenia usiłuje się wykryć i usunąć tego typu cechy. Metody selekcji cech usiłują znaleźć minimalny podzbiór cech, które spełniają następujące kryteria:

- jakość klasyfikacji nie ulegnie znaczącemu zmniejszeniu
- rozkład klas otrzymany przy użyciu tylko wybranych cech jest tak bliski jak to tylko możliwe oryginalnemu rozkładowi tych klas.

Jedną z metod przedstawimy na przykładzie pseudoprób typu bootstrap. Pamiętamy (1.7.1), że do każdej z pseudoprób nie zostaje wybrana średnio około $\frac{1}{3}$ elementów z oryginalnej próby uczącej (zbiór niewybranych elementów z języka angielskiego nazywa się zbiorem *out-of-bag*). Dokonujemy klasyfikacji na zbiorze *out-of-bag* i wyznaczamy liczbę dobrych klasyfikacji. Każdy element *out-of-bag* opisywany jest przez te same cechy. Dokonujemy losowej permutacji wybranej cechy dla wszystkich elementów w zbiorze *out-of-bag*, a następnie sprawdzamy jaka jest teraz jakość klasyfikacji, czy wystąpiła zmiana pomiędzy liczbą obserwacji dobrze sklasyfikowanych po permutacji i przed permutacją. Średnia różnica tych liczb dla wszystkich pseudoprób określa wagę rozpatrywanej cechy i jej wpływ na klasyfikację.

Złożoność i skalowalność

Złożoność obliczeniowa procesu uczenia oraz procesu predykcji jest niezwykle istotna w problemach, gdzie przetwarzane są duże ilości danych. Zwróćmy uwagę, że dane uczące opisywane są dwuwymiarową strukturą, tzn. przez numer obserwacji oraz numer atrybutu. Zatem liczba obserwacji oraz liczba atrybutów w zbiorze uczącym mają wpływ na liczbę iteracji niezbędnych do utworzenia reguły oraz przeprowadzenia predykcji. Zależność liczbowa pomiędzy liczbą iteracji a parametrami wejściowymi algorytmu nazywana jest złożonością obliczeniową algorytmu. W typowych przypadkach atrybuty nominalne powiększają złożoność obliczeniową w stosunku do atrybutów ciągłych / numerycznych. Wyróżnia się również pojęcie złożoności pamięciowej, jednak istnienie nośników potrafiących pomieścić TB danych zmniejsza wagę tej wielkości.

Analiza dynamiki przyrostu danych uczących pozwala na określenie typu zależności pomiędzy sposobem działania klasyfikatora oraz zmianą warunków początkowych. W przypadku harmonijnej zależności mówimy, że proces jest *skalowalny*¹². Własność skalowalności gwarantuje możliwość harmonijnego dostosowywania się systemu w miarę upływu czasu i zwiększania ilości danych, bez konieczności rewolucyjnych zmian projektowych.

¹²Dla algorytmów jest to np. proporcjonalność czasu działania do wielkości danych wejściowych

Rozdział 2

Rodziny klasyfikatorów

2.1. Wprowadzenie

W rozdziale pierwszym sformułowano problem klasyfikacji pod nadzorem jako zadanie wyboru (konstrukcji) klasyfikatora (reguły/funkcji dyskryminacyjnej) na podstawie dostępnej próby uczącej. Podano również metody szacowania jakości predykcji przynależności do klas. W niniejszym rozdziale zastanowimy się nad sytuacjami, gdzie jakość klasyfikacji jest „niezadawalająca”. Zaprezentujemy metody *stabilizacji klasyfikatorów*, u podstaw których leży idea poprawy dokładności działania dowolnej reguły. Techniki te polegają na konstruowaniu wielu wersji jednego klasyfikatora (rodziny klasyfikatorów), przy czym zazwyczaj klasyfikatory z tej rodziny powstają na podstawie pewnych *pseudoprób* utworzonych z oryginalnej próby uczącej.

2.2. Pojęcie rodziny klasyfikatorów

Definicja 2.2.1 *Rodziną klasyfikatorów* nazywamy dowolną rodzinę reguł (hipotez)

$$\mathcal{D} = \left\{ d_k : \mathbb{X} \rightarrow \{1, 2, \dots, g\} \right\}_{k=1,2,\dots,K} \quad (2.1)$$

gdzie $K \geq 2$.

Definicja 2.2.2 *Liczbę głosów* oddanych przez rodzinę klasyfikatorów \mathcal{D} na to, aby obserwację $\mathbf{x} \in \mathbb{X}$ zaklasyfikować do klasy $j \in \{1, 2, \dots, g\}$ określamy jako

$$N_j(\mathbf{x}) := \#\{k : d_k(\mathbf{x}) = j\} \quad (2.2)$$

Definicja 2.2.3 *Klasyfikatorem generowanym* przez rodzinę klasyfikatorów \mathcal{D} nazywamy klasyfikator $d_{\mathcal{D}} : \mathbb{X} \rightarrow \{1, 2, \dots, g\}$ wybrany regułą głosowania

$$d_{\mathcal{D}}(\mathbf{x}) := \arg \max_j N_j(\mathbf{x}) \quad (2.3)$$

Rodzina klasyfikatorów w naturalny sposób generuje regułę poprzez ostateczne zaklasyfikowanie \mathbf{x} do klasy, która była najczęściej wskazywana przez poszczególne reguły z rodziny.

Definicja 2.2.4 *Marginesem klasyfikacji* obserwacji $\mathbf{x} \in \mathbb{X}$ z klasy $y \in \{1, 2, \dots, g\}$ w rodzinie klasyfikatorów \mathcal{D} nazywamy funkcję

$$\text{mg}_{\mathcal{D}}(\mathbf{x}, y) := \frac{1}{K} \sum_{k=1}^K \mathbb{I}_{[d_k(\mathbf{x})=y]} - \max_{j \neq y} \frac{1}{K} \sum_{k=1}^K \mathbb{I}_{[d_k(\mathbf{x})=j]} \quad (2.4)$$

Margines klasyfikacji jest miarą wskazującą o ile średnia liczba głosów oddanych na klasę poprawną przewyższa średnią liczbę głosów oddanych na jakąkolwiek inną klasę. Im większa wartość funkcji marginesu, tym predykcja jest pewniejsza.

Definicja 2.2.5 Jeżeli (\mathbf{x}, y) pochodzą z rozkładu P to *błędem uogólnionym* rodziny klasyfikatorów \mathcal{D} nazywamy prawdopodobieństwo

$$PE^*(\mathcal{D}) := P(\text{mg}_{\mathcal{D}}(\mathbf{x}, y) < 0) \quad (2.5)$$

2.3. Dlaczego rodziny klasyfikatorów?

Rodziny klasyfikatorów posiadają zadziwiająco dobre własności w przypadku, gdy klasyfikator pojedynczy obciążony jest dużym ryzykiem całkowitym, jednak mniejszym niż ryzyko całkowite klasyfikatora losowego. Z klasyfikatorem losowym mamy doczynienia w sytuacji, gdy wybór klasy dokonywany jest w sposób losowy. Takie procesy klasyfikacji nie są zależne od próby uczącej. Prawdopodobieństwo popełnienia błędu (błąd klasyfikacji) przez klasyfikator losowy wyznaczają prawdopodobieństwa a priori przynależności obserwacji do klas. W przypadku dwóch klas z równymi prawdopodobieństwami a priori błąd klasyfikacji wynosi $\frac{1}{2}$. Klasyfikatory niewiele lepsze od losowego wyboru klas nazywamy *słabymi* (słabymi uczniami).

Założmy, że posiadamy K niezależnych klasyfikatorów, gdzie prawdopodobieństwo podjęcia poprawnej decyzji przez każdy z nich wynosi p . Wśród wszystkich decyzji podjętych przez K klasyfikatorów, częstość tych poprawnych wynosi oczywiście p , a wariancja $\frac{p(1-p)}{K}$ (doświadczenie dwumianowe). Ustalając np. prawdopodobieństwo p poprawnej decyzji na poziomie 0,55 oraz zakładając, że dysponujemy 1000 klasyfikatorów, możemy być prawie pewni, że większość z nich dokona poprawnej klasyfikacji. Odpowiednio liczna rodzina niezależnych klasyfikatorów oraz wybór tej klasy, która została wskazana przez większość klasyfikatorów, prawie na pewno jest gwarantem poprawnej decyzji. Takie rozumowanie stoi u podstaw konstrukcji rodzin klasyfikatorów, choć w praktyce dostępne klasyfikatory są od siebie statystycznie zależne (pseudopróby tworzone są na podstawie tej samej próby uczącej). Mimo to rodziny klasyfikatorów posiadają zadziwiająco dobre własności.

Istnieje kilka innych przyczyn skłaniających do konstrukcji rodzin klasyfikatorów oraz do wniosku, że znalezienie pojedynczego klasyfikatora, który działałby tak dobrze jak rodzina, jest trudne. Aby zrozumieć te przyczyny należy rozważyć naturę algorytmów uczących. Algorytmy te przeszukują przestrzeń możliwych reguł w poszukiwaniu reguły najbardziej trafnej. Bardzo ważny jest rozmiar tej przestrzeni, jak i fakt zawierania się w niej dobrego przybliżenia rozważanej zależności. Poniżej przedstawiamy główne przyczyny.

Brak dostatecznej ilości informacji w zbiorze uczącym

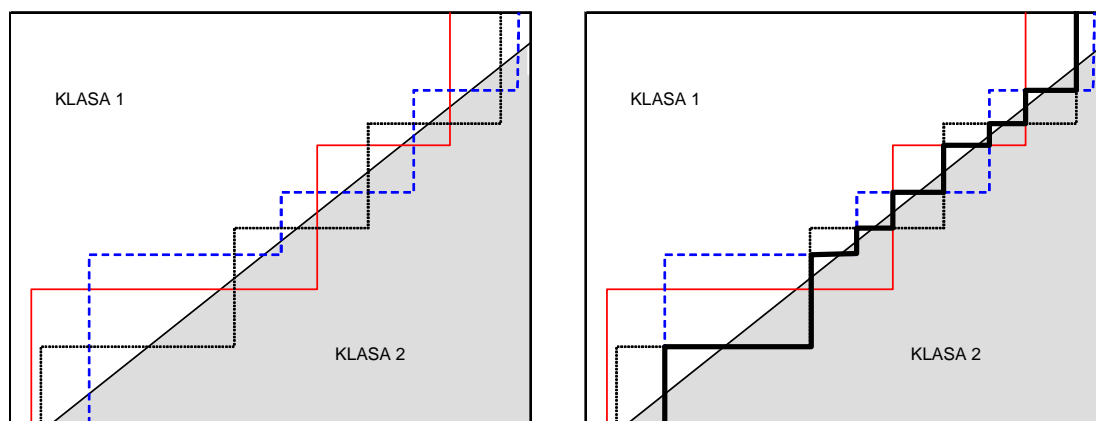
Zbiór uczący może nie dostarczać wystarczającej ilości informacji do wybrania najlepszego pojedynczego klasyfikatora. Większość algorytmów uczących generuje bardzo dużą przestrzeń reguł (hipotez), której rozmiar po wyeliminowaniu tych z dużym błędem, tylko nieznacznie się zmniejsza. Wszystkie pozostałe reguły posiadają wymaganą dokładność klasyfikacji. Możliwe jest jednak zaistnienie różnych powodów, aby niektóre z nich uznać za lepsze (np. hipotezy proste lub z dużym prawdopodobieństwem a priori). Rodziny klasyfikatorów pomagają wybrać właśnie takie hipotezy.

Zrównoważenie niedoskonałości algorytmów uczących

Zdarza się, że algorytmy uczące nie potrafią rozwiązać pewnych trudnych zadań. Przykładem takich zadań są problemy *NP-zupełne* (np.: znajdowanie najmniejszego drzewa decyzyjnego lub też znajdowanie wag dla możliwie najmniejszych sieci neuronowych). Niedoskonałość tych algorytmów objawia się tym, że nie mogą ustalić dokładnego rozwiązania. Najczęściej znajdują pewne reguły, które są bardziej złożone od rozwiązań dokładnych (lub mają mniejsze prawdopodobieństwo a posteriori, są mniej dokładne). Uruchamiając ten sam algorytm uczący na podobnym zbiorze trenującym (np. zaszumionym), otrzymujemy kolejną (różną od poprzedniej) hipotezę (mniej lub bardziej dokładną). Rodziny klasyfikatorów pomagają, zatem zrównoważyć niedoskonałości algorytmów uczących.

Brak dokładnego rozwiązania w przestrzeni hipotez

Przestrzeń hipotez możliwych do utworzenia przez dany algorytm uczący może nie zawierać dokładnej granicy podziału (na rys. 2.1 linia diagonalna), lecz jedynie jej przybliżenia. Skonstruowany klasyfikator na podstawie kombinacji przybliżeń rozważanej zależności może leżeć poza daną przestrzenią (może więc znacznie lepiej przybliżać dokładne rozwiązanie). Dla przykładu algorytmy drzew decyzyjnych przybliżają dokładną granicę podziału przez pewne funkcje - na rys. 2.1 linie schodkowe. Rodziny klasyfikatorów pozwalają na osiągnięcie granicy podziału, która jest znacznie bardziej przybliżona do dokładnej.



Rysunek 2.1: Rozwiązania wskazane przez trzy niezależne klasyfikatory oraz wybrane najlepsze przybliżenie dokładnego rozwiązania.

2.4. Nota historyczna

Pracę nad rodzinami klasyfikatorów rozpoczęto w latach 90-tych XX wieku. Pierwszą i najprostsza rodziną klasyfikatorów jest metoda *bagging*¹. Metoda ta została przedstawiona przez Leo Breimana² w 1996 roku. Kolejna metoda to *boosting*³, która jest ulepszeniem metody

¹Bagging - **bootstrap aggregating**.

²Leo Breiman (1928 - 2005) - wybitny statystyk, Profesor na Uniwersytecie Kalifornijskim w Berkeley. Główne informacje na temat Leo Breimana można znaleźć na stronie <http://www.stat.berkeley.edu/~breiman/>.

³Boosting - z ang. wzmocnienie.

bagging, choć powstawała niezależnie od niej. Pierwszymi, którzy zastanawiali się nad możliwością wzmocnienia „słabego” algorytmu uczącego, którego wyniki działania są nieco lepsze od losowego (random guessing w modelu PAC⁴), byli Michael Kearns⁵ i Leslie Valiant⁶. Natomiast Yoav Freund⁷ i Robert Schapire⁸ w 1995 roku przedstawili algorytm boostingu, który pozwolił już rozwiązać większość praktycznych problemów jakimi były obciążone wcześniejsze jego wersje. Ostatnia technika, którą będziemy się w tej pracy zajmować to *lasy losowe*⁹, zaproponowane przez Leo Breimana.

⁴PAC Learning - Probably Approximately Correct Learning

⁵Michael Kearns - Profesor Nauk Informatycznych na Uniwersytecie Pensylwanii. Strona domowa: <http://www.cis.upenn.edu/~mkearns/>.

⁶Leslie Valiant - Profesor Nauk Informatycznych i Matematyki Stosowanej na Uniwersytecie Harvarda. Strona domowa: <http://people.deas.harvard.edu/~valiant/>.

⁷Yoav Freund - Uniwersytet Kalifornijski w San Diego. Strona domowa: <http://www.cse.ucsd.edu/~yfreund/>.

⁸Robert Schapire - Uniwersytet Princeton. Strona domowa: <http://www.cs.princeton.edu/~schapire/>

⁹Lasy losowe - z ang. random forest.

Rozdział 3

Metoda bagging

3.1. Wprowadzenie

Bagging (*Bootstrap Aggregating*) jest jedną z pierwszych rodzin klasyfikatorów, zaproponowaną przez Breimana w 1996 r. Metoda ta często pozwala poprawić klasyfikację oraz modele regresyjne pod względem stabilności i dokładności. Dodatkowo zastosowanie metody bagging prowadzi do redukcji wariancji pojedynczych klasyfikatorów użytych do konstrukcji rodziny.

3.2. Bootstrap i rodzina bagging

Założmy, że dany mamy zbiór uczący $\mathcal{L} \in \mathbb{L}$ posiadający n elementów oraz klasyfikator $d : \mathbb{X} \times \mathbb{L} \rightarrow \{1, 2, \dots, g\}$. Na podstawie zbioru \mathcal{L} tworzymy K pseudoprób $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_K$. Każda pseudopróba \mathcal{L}_k powstaje w wyniku wylosowania ze zwracaniem n -elementów z wyjściowego zbioru uczącego \mathcal{L} . Zakładamy przy tym, że wylosowanie każdego spośród n elementów jest równoprawdopodobne. Taki sposób generowania pseudoprób nazywamy metodą *bootstrap*. Rozważmy K reguł $d(\cdot, \mathcal{L}_k) : \mathbb{X} \rightarrow \{1, 2, \dots, g\}$. Mówimy, że reguła $d(\cdot, \mathcal{L}_k)$ jest k -tą wersją klasyfikatora d .

W myśl definicji 2.2.1 rodzina

$$\mathcal{B} = \left\{ d(\cdot, \mathcal{L}_k) : \mathbb{X} \rightarrow \{1, 2, \dots, g\} \right\}_{k=1,2,\dots,K} \quad (3.1)$$

jest rodziną klasyfikatorów. Wykorzystując regułę głosowania (def. 2.2.3) otrzymujemy klasyfikator $d_{\mathcal{B}}$ generowany rodziną \mathcal{B} . Klasyfikator $d_{\mathcal{B}}$ nazywamy klasyfikatorem otrzymanym metodą bagging (algorytm 3.1).

Algorytm 3.1 Metoda bagging

```
1 Dane wejściowe:  $\mathcal{L}$ —próba ucząca,  $K$ —liczba iteracji
2   for  $k = 1$  to  $K$  do
3     (a) Z próby uczącej  $\mathcal{L}$  wygeneruj pseudopróbę  $\mathcal{L}_k$ 
4         metodą bootstrap
5     (b) Skonstruuj regułę decyzyjną  $d(\cdot, \mathcal{L}_k)$ 
6   end for
7 Wyjście: Zlicz liczbę głosów  $N_j(\mathbf{x})$ , a następnie oblicz
8          $d_{\mathcal{B}}(\mathbf{x}) = \arg \max_j N_j(\mathbf{x})$ 
```

3.3. Dlaczego bagging działa?

Założmy, że próba ucząca $\mathcal{L} \in \mathbb{L}$ pochodzi z rozkładu P . Niech para (\mathbf{x}, y) niezależna od \mathcal{L} pochodzi również z rozkładu P . Ryzyko całkowite $R_{\mathcal{L}}(d)$ dowolnego klasyfikatora $d(\cdot, \mathcal{L}) : \mathbb{X} \rightarrow \{1, 2, \dots, g\}$, określone w (1.13), wyraża prawdopodobieństwo błędnej klasyfikacji przy ustalonym zbiorze uczącym \mathcal{L} . Możemy zapisać

$$R_{\mathcal{L}}(d) = 1 - P(d(\mathbf{x}, \mathcal{L}) = y) = 1 - \sum_{k=1}^g P(d(\mathbf{x}, \mathcal{L}) = k \mid y = k) P(y = k)$$

Jeżeli $Q(k|\mathbf{x}) = P_{\mathcal{L}}(d(\mathbf{x}, \mathcal{L}) = k)$, to oczekiwane ryzyko całkowite (1.14) przyjmuje postać

$$R^*(d) = 1 - \sum_{k=1}^g \mathbb{E}[Q(k|\mathbf{x}) \mid y = k] P(y = k) = 1 - \sum_{k=1}^g \int Q(k|s) P(k|s) P_{\mathbf{x}}(ds)$$

gdzie $P_{\mathbf{x}}(ds)$ jest gęstością rozkładu P .

Rozważmy klasyfikator

$$d_A(\mathbf{x}) = \arg \max_k Q(k|\mathbf{x}) \quad (3.2)$$

Nietrudno zauważyć, że

$$R^*(d_A) = 1 - \sum_{k=1}^g \int \mathbb{I}_{[\arg \max_l Q(l|\mathbf{x})=k]} P(k|s) P_{\mathbf{x}}(ds)$$

Oznaczając przez \mathcal{C} zbiór obserwacji, dla których klasyfikacja z użyciem reguły d_A daje taki sam wynik jak klasyfikacja wynikająca z rozkładu P (klasyfikacja Bayesowska)

$$\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{X} : \arg \max_k Q(k|\mathbf{x}) = \arg \max_k P(k|\mathbf{x}) \right\}$$

Dla $\mathbf{x} \in \mathcal{C}$ zachodzi

$$\sum_{k=1}^g \mathbb{I}_{[\arg \max_l Q(l|\mathbf{x})=k]} P(k|\mathbf{x}) = \max_k P(k|\mathbf{x})$$

Zatem

$$R^*(d_A) = 1 - \int_{s \in \mathcal{C}} \max_k P(k|s) P_{\mathbf{x}}(ds) - \int_{s \notin \mathcal{C}} \sum_{k=1}^g \mathbb{I}_{[d_A(s)=k]} P(k|s) P_{\mathbf{x}}(ds)$$

Klasyfikator bayesowski \mathbf{b} (def.: 1.4.2) charakteryzuje się najmniejszym z możliwych ryzykiem całkowitym (tw.: 1.4.1).

$$R(\mathbf{b}) = 1 - \int_{s \in \mathbb{X}} \max_k P(k|s) P_{\mathbf{x}}(ds)$$

Jeżeli $\mathbf{x} \in \mathcal{C}$, to suma $\sum_{k=1}^g Q(k|\mathbf{x}) P(k|\mathbf{x})$ może być mniejsza od $\max_k P(k|\mathbf{x})$. Wtedy, nawet gdy $P(\mathcal{C}) \approx 1$, klasyfikator d jest daleki od optymalnego klasyfikatora \mathbf{b}

$$R^*(d) > R^*(\mathbf{b})$$

Jednak klasyfikator d_A (zagregowany) jest prawie optymalny

$$R^*(d) > R^*(d_A) \approx R^*(\mathbf{b})$$

Agregacja może zmienić dobre klasyfikatory w optymalne. Z drugiej strony słabe klasyfikatory mogą zostać zmienione w jeszcze gorsze.

3.3.1. Teoretyczna definicja baggingu

Kolejne zalety metody Bagging przedstawimy na przykładzie bliższym analizie regresji niż problemowi dyskryminacji. Niech $L_i = (X_i, Y_i)$ ($i = 1, \dots, n$), gdzie Y_i jest zmienną objaśnianą, a X_i jest p -wymiarowa zmienna objaśniająca. Bagging definiujemy poprzez

1. Konstrukcję próbek bootstrapowych $L_i^* = (Y_i^*, X_i^*)$ $i = (1, \dots, n)$ zgodnie z empirycznym rozkładem $L_i = (Y_i, X_i)$ ($i = 1, \dots, n$).
2. Obliczenie estymatora bootstrapowego $\hat{\theta}_i^*(\mathbf{x}) = h_n(L_1^*, \dots, L_n^*)(\mathbf{x})$, gdzie $\hat{\theta}_i(\mathbf{x}) = h_n(L_1, \dots, L_n)(\mathbf{x})$.
3. Podanie estymator po baggingu $\hat{\theta}_{n,B}(\mathbf{x}) = \mathbb{E}^*[\hat{\theta}_i^*(\mathbf{x})]$.

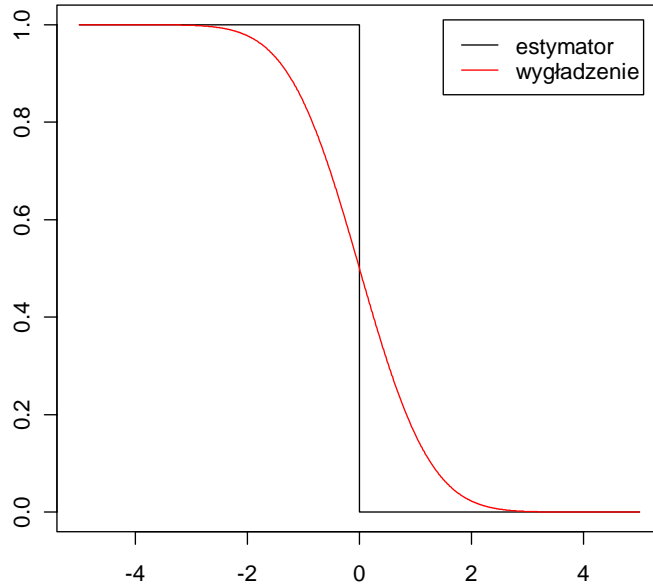
Definicja 3.3.1 Statystykę $\hat{\theta}_i(\mathbf{x}) = h_n(L_1, \dots, L_n)(\mathbf{x})$ nazywamy stabilną na \mathbf{x} , jeśli $\hat{\theta}_i(\mathbf{x}) = \theta(\mathbf{x}) + o_p(1)$ ($n \rightarrow \infty$) dla ustalonego $\theta(\mathbf{x})$.

Niech Y_1, \dots, Y_n będzie prostą próbą losową (iid).

Rozważmy:

$$\hat{\theta}_n(\mathbf{x}) = \mathbb{I}_{[\bar{Y}_n \leq \mathbf{x}]} \quad \mathbf{x} \in \mathbb{R}, \quad \text{gdzie} \quad \bar{Y}_n = \frac{1}{n} \sum_{i=1}^n Y_i.$$

Ograniczymy się jedynie do przypadku, gdy \mathbf{x} jest ustalony. Dalsze rozważania pokażą, że bagging powoduje „złagodzenie” progowego charakteru średniej \bar{Y}_n na ustalonym \mathbf{x} .



Rysunek 3.1: Dla ustalonego $\mathbf{x} = 0$, funkcja $\hat{\theta}_n(\mathbf{x})$ oraz jej zmiana („wygładzanie”) po baggingu.

Dla \mathbf{x} znajdującego się w sąsiedztwie parametru μ

$$\mathbf{x} = \mathbf{x}_n(c) = \mu + \frac{c\delta}{\sqrt{n}}$$

możemy zapisać:

$$\hat{\theta}_n(\mathbf{x}_n(c)) = \mathbb{I}_{[\bar{Y}_n \leq \mathbf{x}_n(c)]} = \mathbb{I}_{[\bar{Y}_n \leq \mu + \frac{c\delta}{\sqrt{n}}]} = \mathbb{I}_{[\frac{(\bar{Y}_n - \mu)\sqrt{n}}{\delta} \leq c]} = \mathbb{I}_{[Z \leq c]} \quad Z \sim N(0, 1).$$

Natomiast:

$$(*) \quad \mathbb{E}(\hat{\theta}_n(\mathbf{x}_n(c))) \xrightarrow[n \rightarrow \infty]{d} \mathbb{E}(\mathbb{I}_{[Z \leq c]}) = P(Z \leq c) = \Phi(c)$$

$$\text{Var}(\hat{\theta}_n(\mathbf{x}_n(c))) \xrightarrow[n \rightarrow \infty]{d} \mathbb{E}(\mathbb{I}_{[Z \leq c]})^2 - (\mathbb{E}(\mathbb{I}_{[Z \leq c]}))^2 = \Phi(c)(1 - \Phi(c)),$$

gdzie Φ dystrybuanta standardowego rozkładu normalnego.

Dopóki wariancja nie zbiega do zera, to $\hat{\theta}_n(\mathbf{x}_n(c))$ jest niestabilny w sensie definicji 3.3.1 (klasyfikator przyjmuje wartości 0 i 1 z dodatnim prawdopodobieństwem nawet przy $n \rightarrow \infty$). Estymator baggingowy przedstawia się w następujący sposób:

$$\begin{aligned} \hat{\theta}_{n;B}(\mathbf{x}_n(c)) &= \mathbb{E}^* \left(\mathbb{I}_{[\bar{Y}_n^* \leq \mathbf{x}_n(c)]} \right) = \mathbb{E}^* \left(\mathbb{I}_{\left[\frac{\sqrt{n}(\bar{Y}_n^* - \bar{Y}_n)}{\delta} \leq \frac{\sqrt{n}(\mathbf{x}_n(c) - \bar{Y}_n)}{\delta} \right]} \right) = \\ &= \Phi \left(\frac{\sqrt{n}(\mathbf{x}_n(c) - \bar{Y}_n)}{\delta} \right) + o_p(1) \approx \Phi(c - Z) \quad Z \sim N(0, 1). \end{aligned}$$

Bardzo pouczającym i obrazującym działaniem baggingu jest przypadek, gdy $\mathbf{x} = \mathbf{x}_n(0) = \mu$. Jest to najbardziej niestabilne położenie, $\text{Var}(\hat{\theta}_n(\mathbf{x}))$ osiąga wartość maksymalną. Natomiast:

$$\hat{\theta}_{n;B}(\mathbf{x}_n(0)) \xrightarrow[n \rightarrow \infty]{d} \Phi(-Z) = U$$

Przypomnijmy, że jeżeli zmienna losowa ma rozkład o ciągłej i ściśle rosnącej dystrybuancie F , to zmienna losowa $R = F(X)$ ma rozkład równomierny na $(0, 1)$. Zatem $U \sim U([0, 1])$.

$$\mathbb{E}(\hat{\theta}_{n;B}(\mathbf{x}_n(0))) \xrightarrow[n \rightarrow \infty]{d} \mathbb{E}(U) = \frac{1}{2}$$

$$\text{Var}(\hat{\theta}_{n;B}(\mathbf{x}_n(0))) \xrightarrow[n \rightarrow \infty]{d} \text{Var}(U) = \frac{1}{4}.$$

Porównując z (*):

$$\mathbb{E}(\hat{\theta}_{n;B}(\mathbf{x}_n(0))) \xrightarrow[n \rightarrow \infty]{d} \Phi(0) = \frac{1}{2}$$

$$\text{Var}(\hat{\theta}_{n;B}(\mathbf{x}_n(0))) \xrightarrow[n \rightarrow \infty]{d} \Phi(c)(1 - \Phi(c)) = \frac{1}{4}$$

można zauważyć, że bagging redukuje wariancję. Poprawia w ten sposób jakość estymacji. Rozważmy teraz przypadek bardziej ogólny:

$$\hat{\theta}_n(\mathbf{x}) = \mathbb{I}_{[\hat{d}_n \leq x]}, \quad \mathbf{x} \in R$$

przy następujących założeniach:

$$b_n(\hat{d}_n - d^0) \xrightarrow[n \rightarrow \infty]{d} N(0, \delta_\infty^2)$$

$$\sup \left| P^*[b_n(\hat{d}_n^* - \hat{d}_n) \leq v] - \Phi\left(\frac{v}{\delta_\infty}\right) \right| = o_p(1),$$

gdzie $0 < \delta_\infty^2 < \infty$, $(b_n)_{n \in \mathbb{N}}$ - ciąg rosnący, \hat{d}_n^* - „bootstrapowy” estymator. Podsumowując powyższe założenia otrzymujemy:

$$\hat{\theta}_n(\mathbf{x}_n(c)) \xrightarrow[n \rightarrow \infty]{d} g(Z) = 1_{[Z \leq c]}$$

$$\hat{\theta}_{n;B}(\mathbf{x}_n(c)) \xrightarrow[n \rightarrow \infty]{d} g_B(Z) = \Phi(c - Z), \quad \text{gdzie } Z \sim N(0, 1).$$

Wniosek 3.3.1

1. $\lim_{n \rightarrow \infty} \mathbb{E}[\hat{\theta}_n(\mathbf{x}_n(c))] = \Phi(c), \quad \lim_{n \rightarrow \infty} \text{Var}(\hat{\theta}_n(\mathbf{x}_n(c))) = \Phi(c)(1 - \Phi(c)).$
2. $\lim_{n \rightarrow \infty} \mathbb{E}[\hat{\theta}_{n;B}(\mathbf{x}_n(c))] = \Phi * \varphi(c), \quad \lim_{n \rightarrow \infty} \text{Var}(\hat{\theta}_{n;B}(\mathbf{x}_n(c))) = \Phi * \varphi(c) - (\Phi * \varphi(c))^2,$ gdzie $f * g(\cdot) = \int_{\mathbb{R}} f(\cdot - y) \times g(y) dy$ i $\varphi(\cdot)$ - standardowa gęstość rozkładu normalnego.

Rozdział 4

Metoda boosting

4.1. Wprowadzenie

W poprzednim rozdziale przedstawiono pierwszą i najprostszą metodę łączenia klasyfikatorów. Poniżej zamieszczamy opis kolejnej metody, bardziej złożonej choć w konstrukcji bardzo podobnej, z reguły dającej lepsze wyniki klasyfikacji. Mowa jest o metodzie *boosting*. Technika ta bardzo często z dobrymi efektami wykorzystywana jest do rozpoznawania tekstu.

4.2. Algorytm AdaBoost

Założmy, że dany mamy zbiór uczący $\mathcal{L} \in \mathbb{L}$ posiadający n elementów oraz klasyfikator $d : \mathbb{X} \times \mathbb{L} \rightarrow \{1, 2, \dots, g\}$. Na podstawie zbioru \mathcal{L} tworzymy K pseudoprób $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_K$. Każda pseudopróba \mathcal{L}_k powstaje w wyniku losowania ze zwracaniem n -elementów z wyjściowego zbioru uczącego \mathcal{L} , przy czym rozkład prawdopodobieństwa, zgodnie z którym dokonuje się losowania elementów do danej pseudopróby zmienia się w kolejnych krokach algorytmu. W k -tym kroku procedury tworzona jest pseudopróba \mathcal{L}_k , a na jej podstawie konstruowana jest k -ta wersja $d(\cdot, \mathcal{L}_k) : \mathbb{X} \rightarrow \{1, 2, \dots, g\}$ klasyfikatora d . Następnie sprawdzana jest jakości k -tego klasyfikatora na oryginalnej próbie uczącej \mathcal{L} . Losowanie elementów do pseudopróby \mathcal{L}_1 odbywa się zgodnie z rozkładem jednostajnym. W kolejnych krokach rozkład prawdopodobieństwa, według którego losowane są elementy, jest adaptacyjnie zmieniany. Jeżeli w k -tym kroku i -ta obserwacja była losowana z prawdopodobieństwem w_i oraz ta obserwacja została błędnie zaklasyfikowana przez k -tą wersję klasyfikatora, to w kroku $k + 1$ prawdopodobieństwo wylosowania i -tej obserwacji do pseudopróby \mathcal{L}_{k+1} zostaje zwiększone. W konsekwencji prawdopodobieństwa losowania obserwacji zaklasyfikowanych poprawnie są odpowiednio zmniejszone.

W myśl definicji 2.2.1 rodzina

$$\mathcal{A} = \left\{ d(\cdot, \mathcal{L}_k) : \mathbb{X} \rightarrow \{1, 2, \dots, g\} \right\}_{k=1,2,\dots,K} \quad (4.1)$$

jest rodziną klasyfikatorów. Wykorzystując zmodyfikowaną regułę głosowania (opis w algorytmie 4.1) otrzymujemy klasyfikator $d_{\mathcal{A}}$ generowany rodziną \mathcal{A} . Klasyfikator $d_{\mathcal{A}}$ nazywamy klasyfikatorem otrzymanym metodą boosting.

Zasadę działania boostingu przedstawimy na przykładzie algorytmu AdaBoost (alg. 4.1). Algorytm ten został wprowadzony w 1995 r. przez *Freund'a* i *Schapire'a*.

Ogólna idea algorytmu AdaBoost

W pierwszym kroku algorytm AdaBoost inicjalizuje wagi dla każdej obserwacji, nadając im tę samą wartość $\frac{1}{n}$. Dalej następuje iteracyjne:

1. normalizowanie (renormalizowanie) wag;
2. wybór pseudopróby na podstawie wag;
3. wyznaczanie wersji klasyfikatora;
4. wyznaczanie błędu wersji klasyfikatora;
5. wyznaczanie nowych wag na podstawie błędu wersji klasyfikatora.

Proces iteracji kończy się, gdy błąd wersji klasyfikatora przekroczy wartość $\frac{1}{2}$ lub gdy liczba iteracji osiągnie ustaloną maksymalną liczbę iteracji. Szczegółowy opis AdaBoost zawiera algorytm 4.1.

Algorytm 4.1 Metoda AdaBoost

```

1  Dane wejściowe:  $\mathcal{L}$ —próba ucząca,  $K$ —liczba iteracji
2      for all  $i$  : zainicjuj wagi  $w_{1i} := \frac{1}{n}$ 
3      for  $k = 1$  to  $K$  do
4          for all  $i$  : znormalizuj wagi  $p_{ki} := w_{ki} / \sum_i w_{ki}$ 
5          wylosuj pseudopróbę  $\mathcal{L}_k$  z uwzględnieniem wag
6          wyznacz  $k$ -tą wersję klasyfikatora  $d_k := d(\cdot, \mathcal{L}_k)$ 
7          wyznacz błąd  $\varepsilon_k = \sum_i p_{ki} \mathbb{I}_{[h_k(\mathbf{x}_i) \neq y_i]}$ 
8          if  $\varepsilon_k > \frac{1}{2}$  then do
9               $K := k - 1$ 
10             Wyjście
11         end if
12          $\beta_k := \varepsilon_k / (1 - \varepsilon_k)$ 
13         for all  $i$  : zaktualizuj wagi  $w_{k+1}(i) := w_k(i) \beta_k^{1 - \mathbb{I}_{[h_k(\mathbf{x}_i) \neq y_i]}}$ 
14     end for
15 Wyjście:  $d_A(\mathbf{x}) = \arg \max_j \sum_{k=1}^K (\log \frac{1}{\beta_k}) \mathbb{I}_{[d_k(\mathbf{x})=j]}$ 

```

4.3. AdaBoost jako addytywny model regresji logistycznej

Algorytm AdaBoost przedstawia przybliżony, iteracyjny sposób rozwiązania zadania dyskryminacji logistycznej (sekcja 1.6). Rozważając model logistyczny przy problemie dwóch klas kodowanych jako $\{-1, 1\}$ otrzymujemy funkcję logitową:

$$\ln \frac{P(y = 1|\mathbf{x})}{1 - P(y = -1|\mathbf{x})} = \ln \frac{P(y = 1|\mathbf{x})}{P(y = -1|\mathbf{x})}.$$

Ponieważ funkcja logitowa jest funkcją obserwacji \mathbf{x} , zatem możemy przyjąć

$$\ln \frac{P(y = 1|\mathbf{x})}{P(y = -1|\mathbf{x})} = \tilde{F}(\mathbf{x}) \quad (4.2)$$

gdzie $\tilde{F}(\cdot)$ jest pewną funkcją. Rozwikłując model 4.2 otrzymujemy

$$P(y = 1|\mathbf{x}) = \frac{e^{\tilde{F}(\mathbf{x})}}{1 + e^{\tilde{F}(\mathbf{x})}}$$

Wykażemy, że przy konstrukcji algorytmu AdaBoost założenie, że $\tilde{F}(\cdot)$ jest funkcją liniową nie jest konieczne.

Lemat 4.3.1 Niech

$$J(F) = \mathbb{E}[e^{-yF(\mathbf{x})}] \quad (4.3)$$

gdzie $F(\cdot)$ jest funkcją rzeczywistą i operator \mathbb{E} oznacza wartość oczekiwaną względem łącznego rozkładu wektora (\mathbf{x}, y) . Funkcja $J(F)$ osiąga minimum dla

$$F(\mathbf{x}) = \frac{1}{2} \ln \frac{P(y = 1|\mathbf{x})}{P(y = -1|\mathbf{x})} \quad (4.4)$$

Zatem

$$P(y = 1|\mathbf{x}) = \frac{e^{2F(\mathbf{x})}}{1 + e^{2F(\mathbf{x})}} = \frac{e^{F(\mathbf{x})}}{e^{-F(\mathbf{x})} + e^{F(\mathbf{x})}} \quad (4.5)$$

oraz

$$P(y = -1|\mathbf{x}) = \frac{1}{1 + e^{2F(\mathbf{x})}} = \frac{e^{-F(\mathbf{x})}}{e^{-F(\mathbf{x})} + e^{F(\mathbf{x})}}. \quad (4.6)$$

Dowód: Należy dokonać minimalizacji warunkowej wartości oczekiwanej funkcji $J(\cdot)$ pod warunkiem \mathbf{x} :

$$\mathbb{E}(e^{-yF(\mathbf{x})}|\mathbf{x}) = P(y = 1|\mathbf{x})e^{-F(\mathbf{x})} + P(y = -1|\mathbf{x})e^{F(\mathbf{x})}.$$

Obliczmy pochodną

$$\frac{\partial \mathbb{E}(e^{-yF(\mathbf{x})}|\mathbf{x})}{\partial F(\mathbf{x})} = -P(y = 1|\mathbf{x})e^{-F(\mathbf{x})} + P(y = -1|\mathbf{x})e^{F(\mathbf{x})}$$

Następnie

$$\frac{\partial \mathbb{E}(e^{-yF(\mathbf{x})}|\mathbf{x})}{\partial F(\mathbf{x})} = 0$$

$$-P(y = 1|\mathbf{x})e^{-F(\mathbf{x})} + P(y = -1|\mathbf{x})e^{F(\mathbf{x})} = 0$$

stąd

$$F(\mathbf{x}) = \frac{1}{2} \ln \frac{P(y = 1|\mathbf{x})}{P(y = -1|\mathbf{x})}.$$

■

Założmy, że $F(\mathbf{x})$ jest przybliżonym rozwiązaniem zadania minimalizacji 4.3 otrzymanym iteracyjnie. Szukamy kolejnego rozwiązania postaci $F(\mathbf{x}) + cf(\mathbf{x})$. Ustalając współczynnik c i punkt \mathbf{x} przybliżamy wartości $J(F(\mathbf{x}) + cf(\mathbf{x}))$, wykorzystując rozwinięcia w szereg Taylora wokół punktu $f(\mathbf{x}) = 0$ z resztą trzeciego rzędu. Dodatkowo zakładamy, że $f(\mathbf{x}) \in \{-1, 1\}$. Podane założenie o funkcji $f(x)$ doprowadzi do otrzymania teoretycznego odpowiednika algorytmu AdaBoost.

Pomijając resztę, czyli zgadzając się na przybliżenie kwadratowe, możemy zapisać

$$J(F(\mathbf{x}) + cf(\mathbf{x})) = \mathbb{E}(e^{-y(F(\mathbf{x}) + cf(\mathbf{x}))}) \approx \mathbb{E}(e^{-yF(\mathbf{x})}(1 - cyf(\mathbf{x}) + \frac{1}{2}c^2y^2f(\mathbf{x})^2)) =$$

$$= \mathbb{E}\left(e^{-yF(\mathbf{x})}\left(1 - ycf(\mathbf{x}) + \frac{1}{2}c^2\right)\right),$$

ponieważ $y^2 = 1$ oraz $f(\mathbf{x})^2 = 1$. Minimalizując powyższe wyrażenie w punkcie \mathbf{x} ze względu na funkcję $f(\mathbf{x}) \in \{-1, 1\}$, otrzymujemy

$$f(\mathbf{x}) = \arg \min_f \mathbb{E}_{\mathbf{w}}\left(1 - ycf(\mathbf{x}) + \frac{1}{2}c^2 \mid \mathbf{x}\right), \quad (4.7)$$

gdzie

$$\mathbb{E}_{\mathbf{w}}[g(\mathbf{x}, y) \mid \mathbf{x}] \equiv \frac{\mathbb{E}[\mathbf{w}(\mathbf{x}, y)g(\mathbf{x}, y) \mid \mathbf{x}]}{\mathbb{E}[\mathbf{w}(\mathbf{x}, y) \mid \mathbf{x}]}$$

oraz $\mathbf{w} = \mathbf{w}(\mathbf{x}, y) = e^{-yF(\mathbf{x})}$.

Dla dowolnej wartości $c > 0$ minimalizacja wyrażenia 4.7 jest równoważne maksymalizacji

$$\arg \max_f \mathbb{E}_{\mathbf{w}}[yf(\mathbf{x}) \mid \mathbf{x}]. \quad (4.8)$$

Rozwiązaniem zadania 4.8 jest funkcja:

$$f(\mathbf{x}) = \begin{cases} 1 & \text{gd}y \mathbb{E}_{\mathbf{w}}(y \mid \mathbf{x}) = P_{\mathbf{w}}(y = 1 \mid \mathbf{x}) - P_{\mathbf{w}}(y = -1 \mid \mathbf{x}) > 0 \\ -1 & \text{w.p.p.} \end{cases} \quad (4.9)$$

Zauważmy, że

$$-\mathbb{E}_{\mathbf{w}}[yf(\mathbf{x})] = \mathbb{E}_{\mathbf{w}}[y - f(\mathbf{x})]^2/2 - 1, \quad (4.10)$$

ponieważ $y^2 = f(\mathbf{x})^2 = 1$.

Z równości 4.10 wynika, że minimalizacja kwadratowego przybliżenia funkcji $J(F)$ ze względu na $f(\mathbf{x})$ sprowadza się do wyboru wartości $f(\mathbf{x}) \in \{-1, 1\}$, opartego na metodzie ważonych najmniejszych kwadratów.

Mając minimum ze względu na $f(\mathbf{x})$, przy czym $f(\mathbf{x}) \in \{-1, 1\}$, prawdziwe dla każdej wartości $c > 0$, możemy wyznaczyć najlepszą wartość c , minimalizując wyrażenie $J(F(\mathbf{x}) + cf(\mathbf{x}))$ ze względu na c :

$$c = \arg \min_c \mathbb{E}_{\mathbf{w}} e^{-cyf(\mathbf{x})} = \frac{1}{2} \ln \frac{1 - \text{err}}{\text{err}}$$

gdzie

$$\text{err} = \mathbb{E}_{\mathbf{w}} \left[\mathbb{I}_{[y \neq f(\mathbf{x})]} \right].$$

Ostatecznie algorytm minimalizacji funkcji 4.3 przyjmuje postać:

$$F(\mathbf{x}) \leftarrow F(\mathbf{x}) + \frac{1}{2} \ln \left(\frac{1 - \text{err}}{\text{err}} \right) f(\mathbf{x}).$$

W kolejnym kroku składnik $cf(\mathbf{x})$ musi zostać uwzględniony przy aktualizacji wag,

$$\mathbf{w}(\mathbf{x}, y) \leftarrow \mathbf{w}(\mathbf{x}, y) e^{-cf(\mathbf{x})y}.$$

Ponieważ $-yf(\mathbf{x}) = 2 \times \mathbb{I}_{[y \neq f(\mathbf{x})]} - 1$, aktualizacji wag możemy nadać postać

$$\mathbf{w}(\mathbf{x}, y) \leftarrow \mathbf{w}(\mathbf{x}, y) \exp \left(\ln \left(\frac{1 - \text{err}}{\text{err}} \right) \mathbb{I}_{[y \neq f(\mathbf{x})]} \right).$$

Otrzymujemy iteracyjne przybliżenie rozwiązania zadania minimalizacji 4.3, a zatem algorytmiczna droga poszukiwania modelu 4.2, dokładnie odpowiada krokom algorytmu AdaBoost.

4.4. Błąd klasyfikacji algorytmem AdaBoost

Główną zaletą algorytmu AdaBoost jest zdolność do redukowania błędu uczenia. *Schapire* i *Singer* pokazali, że błąd uczenia ostatecznej reguły dyskryminacyjnej $d_{\mathcal{A}}(\mathbf{x})$, otrzymanej jako kombinacja wersji klasyfikatorów podczas głosowania większościowego, jest ograniczony. Jeżeli każda wersja $d(\cdot, \mathcal{L}_k)$ klasyfikatora $d_{\mathcal{A}}(\mathbf{x})$ jest nieznacznie lepsza niż klasyfikator losowy, wtedy błąd uczenia maleje wykładniczo. To powoduje, że technika boostingu potrafi zmienić słaby algorytm uczący w niezwykle efektywny. *Freund* i *Schapire* pokazali natomiast, że uogólniony błąd końcowej reguły $d_{\mathcal{A}}(\mathbf{x})$ ma pewne ograniczenie górne, które zależy od:

1. błędu klasyfikacji,
2. wymiaru danych uczących m ;
3. wymiaru \mathbf{VC}^1 przestrzeni hipotez d ;
4. liczby iteracji K .

Ograniczenie to ma następującą postać:

$$P\left(d_{\mathcal{A}}(\mathbf{x}) \neq y\right) + \tilde{O}\left(\sqrt{\frac{Kd}{m}}\right)$$

gdzie $P(\cdot)$ jest prawdopodobieństwem empirycznym dla zbioru uczącego.

Alternatywną analizę błędu zaproponował *Shapire*. Posłużył się on pojęciem marginesu (swobody). Definicję marginesu *Shapire*'a podajemy dla przypadku, gdy zbiór klas jest dwuelementowy.

Definicja 4.4.1 Dla obserwacji \mathbf{x} z klasy $y \in \{-1, 1\}$ marginesem (swobodą) nazywamy wartość

$$\text{mg}_S(\mathbf{x}, y) = \frac{y \sum_k \alpha_k d_k(\mathbf{x})}{\sum_k \alpha_k} \quad (4.11)$$

gdzie:

$$\alpha_k = \frac{1}{2} \log \left(\frac{1}{\beta_k} \right)$$

Ograniczenie przedstawione przez *Shapire*'a dla dwóch klas $\{-1, 1\}$ ma postać:

$$P\left(\text{mg}_S(\mathbf{x}, y) \leq \theta\right) + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right) \quad \forall \theta > 0$$

Ograniczenie to nie zależy od liczby iteracji K . Udowodniono również, że boosting szczególnie koncentruje się na obiektach o niewielkiej swobodzie.

Ciekawą cechą metod bagging i boosting jest to, że oszacowanie prawdopodobieństwa błędnej klasyfikacji przez rodzinę jest często niezwykle „gładką” funkcją liczby użytych klasyfikatorów (np. drzew klasyfikacyjnych - sekcja 5.2, ciekawe przykłady dla drzewa klasyfikacyjnych

¹Wymiar VC (Vapnika-Czervononenkisa) przestrzeni hipotez \mathcal{H} , oznaczany przez $VC(\mathcal{H})$, jest definiowany jako maksymalna wartość d taka, że w dziedzinie \mathbf{X} istnieje d różnych przykładów, które mogą być oznaczone jako pozytywne lub negatywne przez hipotezy z przestrzeni \mathcal{H} na wszystkie 2^d możliwych sposobów. Jeśli jest to możliwe dla dowolnie wielu różnych elementów dziedziny, to $VC(\mathcal{H}) = \infty$.

zamieszczone są w sekcji 6.3 niniejszej pracy). Krzywa najczęściej początkowo maleje i następnie stabilizuje się, tzn. od pewnego punktu nie ma istotnej poprawy klasyfikacji wraz ze wzrostem liczby klasyfikatorów w rodzinie. Pojawiło się nawet przekonanie, że rodziny klasyfikatorów nie podlegają zjawisku przetrenowania (nadmiernego dopasowania do zbioru uczącego - sekcja 5.2.4). Doświadczenia dowodzą, że boosting na ogół lepiej się spisuje od metody bagging. W przypadku boostingu pojawiło się również podejrzenie, że wraz ze wzrostem liczby klasyfikatorów prawdopodobieństwo błędnej klasyfikacji dąży do minimum równego ryzyku bayesowskiemu. Powyższe podejrzenie okazało się nieprawdziwe. Jednak Breimanowi udało się udowodnić, że boosting jest asymptotycznie optymalny dla próby o nieskończonej liczności. Powstała również idea regularyzacji estymatorów i klasyfikatorów, która prowadzi do asymptotycznej optymalności ich metod.

4.4.1. Asymptotyczna optymalność algorytmu AdaBoost

W 2004 roku Leo Breiman opublikował w [8] dowód bardzo ciekawego i ważnego faktu, że boosting jest asymptotycznie bayesowski jeśli próba ucząca jest nieskończenie liczna (dowód opiera się na dokładnych wartościach oczekiwanych oraz dokładnych rozkładach).

Breiman postawił następujące założenia:

- Zbiór obserwacji \mathbb{X} jest skończonym p -wymiarowym prostokątem w przestrzeni Euklidesowej \mathbb{R}^p , tzn.

$$\mathbb{X} = \left\{ \left(x^{(1)}, x^{(2)}, \dots, x^{(p)} \right) = \mathbf{x} \in \mathbb{R}^p : a_i \leq x^{(i)} \leq b_i, i = 1, 2, \dots, p \right\}$$

- Rozkład wektora losowego \mathbf{x} jest znany i dany przez $P(d\mathbf{x})$, zakładamy dodatkowo, że rozkład \mathbf{x} jest absolutnie ciągły według p -wymiarowej miary Lebesgue'a i piszemy $P(d\mathbf{x}) = f(\mathbf{x})d\mathbf{x}$.
- Przez $y \in \{-1, 1\}$ oznaczamy etykietę klasy do której należy wektora \mathbf{x} , rozkład y dany jest przez $P(y|\mathbf{x})$ - zakładamy, że jest znany.
- Elementami rodziny boosting będą drzewa klasyfikacyjne posiadające T liście (każde drzewo w rodzinie ma tyle samo liści). Testy w drzewach mają postać testów nierównościowych ($x \leq w$). Do każdego liścia przyporządkowuje się klasę albo -1 albo 1 . Takie drzewa nazywać będziemy ± 1 -drzewami.

Pomysł Breimana polegał na rozważeniu przestrzeni $L^2(\mathbb{X}, P)$ funkcji rzeczywistych na p -wymiarowym prostokącie \mathbb{X} , całkowalnych z 2-gą potęgą względem miary $P(d\mathbf{x})$ (prawdopodobieństwa P). Przestrzeń $L^2(\mathbb{X}, P)$ jest oczywiście przestrzenią Hilberta [11], tzn. jest przestrzenią unitarną (z iloczynem skalarnym) oraz zupełną (względem odległości wyznaczonej przez normę w L^2). Breiman następnie konstruuje układ klasyfikatorów (funkcji w L^2), który jest zupełny. Zupełność takiego układu w przestrzeni Hilberta L^2 gwarantuje możliwość reprezentacji dowolnej funkcji w L^2 przez kombinację liniową funkcji z układu bądź przez granicę ciągu takich kombinacji liniowych (bezpośredni wniosek z zupełności przestrzeni oraz zupełności układu). Przy założeniu, że regułą bayesa otrzymujemy hipotezę (funkcję) należącą do L^2 , powyższe implikuje możliwość osiągnięcia ryzyka bayesowskiego przez kombinację liniową klasyfikatorów z danego układu lub przez ciąg takich kombinacji.

Definicja 4.4.2 Układ (zbiór) funkcji $F \subset L^2(\mathbb{X}, P)$ nazywamy zupełnym jeżeli domknięcie w L^2 zbioru skończonych kombinacji liniowych elementów z F , oznaczane przez $\bar{c}(F)$, równe jest całej przestrzeni L^2 .

Przypomnijmy, że iloczyn skalarny w $L^2(\mathbb{X}, P)$ przyjmuje postać

$$(f|g) := \int_{\mathbb{X}} f(\mathbf{x})g(\mathbf{x})P(d\mathbf{x}) \quad (4.12)$$

Lemat 4.4.1 Układ (zbiór) funkcji $F \subset L^2(\mathbb{X}, P)$ jest zupełny wtedy i tylko wtedy, gdy z warunku $(g|f) = 0$ wynika, że $f = 0$ dla wszystkich $f \in F$.

Powyższy lemat mówi, że układ funkcji jest zupełny wtedy i tylko wtedy, gdy nie istnieje w $L^2(\mathbb{X}, P)$ taka niezerowa funkcja g , że $(g|f) = 0$ dla wszystkich $f \in F$.

Lemat 4.4.2 Układ (zbiór) funkcji $F \subset L^2(\mathbb{X}, P)$ jest zupełny jeżeli $\bar{c}(F)$ zawiera wszystkie funkcje charakterystyczne p -wymiarowych podprostokątów w \mathbb{X} .

Definicja 4.4.3 ± 1 -drzewem nazywamy taką funkcję (hipotezę) $t : \mathbb{X} \rightarrow \{-1, 1\}$ reprezentowaną przez drzewo o T liściach, że dla każdego $\mathbf{x} \in \mathbb{X}$ funkcja $t(\mathbf{x})$ przyjmuje wartość albo -1 albo 1 .

Lemat 4.4.3 Jeżeli $T > p$ to klasa ± 1 -drzew jest zupełna w $L^2(\mathbb{X}, P)$.

Dowód: Pokażemy, że funkcja charakterystyczna dowolnego p -wymiarowego podprostokąta w \mathbb{X} może być przedstawiona jako skończona kombinacja liniowa elementów z klasy ± 1 -drzew. Rozważmy podprostokąt R_0 prostokąta \mathbb{X} ($R_0 \subseteq \mathbb{X}$). Możemy zapisać:

$$R_0 = \{r_1 < x^{(1)} \leq s_1, r_2 < x^{(2)} \leq s_2, \dots, r_p < x^{(p)} \leq s_p\} \quad (4.13)$$

Drzewo posiadające T węzłów wykorzystuje $T - 1 \geq p$ podziałów. Przy testach nierównościowych otrzymujemy zatem zbiór

$$R_1 = \{x^{(1)} \leq s_1, x^{(2)} \leq s_2, \dots, x^{(p)} \leq s_p\} \quad (4.14)$$

przyjmując, że drzewo t_{R_1} zwraca wartość 1 na zbiorze R_1 (posiada przynajmniej jeden liść z etykietą klasy 1). Niech t_{R_2} oznacza drzewo o identycznych liściach jak drzewo T_{R_1} , poza R_1 o przeciwnych w stosunku do T_{R_1} etykietach klas. Wtedy $\frac{1}{2}t_{R_1} + \frac{1}{2}t_{R_2}$ jest funkcją charakterystyczną zbioru R_1 . Powtarzając powyższe otrzymujemy funkcję charakterystyczną zbioru

$$R_2 = \{x^{(1)} \leq r_1, x^{(2)} \leq s_2, \dots, x^{(p)} \leq s_p\} \quad (4.15)$$

Odejmując od funkcji charakterystycznej zbioru R_1 otrzymujemy funkcję charakterystyczną zbioru

$$R_3 = \{r_1 < x^{(1)} \leq s_1, x^{(2)} \leq s_2, \dots, x^{(p)} \leq s_p\} \quad (4.16)$$

Powtarzanie procesu prowadzi do funkcji charakterystycznej zbioru R_0 . ■

Zupełność klasy ± 1 drzew jest bardzo ciekawą własnością procesu klasyfikacji przy wykorzystaniu rodzin klasyfikatorów. Załóżmy, że hipoteza minimalizująca ryzyko całkowite klasyfikatora (pewnego) jest elementem przestrzeni $L^2(\mathbb{X}, P)$.

Twierdzenie 4.4.1 Istnieje liniowa kombinacja ± 1 -drzew w prostokącie \mathbb{X} , która zbiega w L^2 do dowolnej hipotezy minimalizującej ryzyko całkowite.

Ryzyko całkowite klasyfikatora ϕ możemy zapisać jako

$$R^*(\phi) = P_{y,\mathbf{x}}(y \neq \phi(\mathbf{x}))$$

Zakładając, że

$$\phi(\mathbf{x}) := \begin{cases} -1 & \text{gdy } \phi(\mathbf{x}) > 0 \\ 1 & \text{gdy } \phi(\mathbf{x}) \leq 0 \end{cases}$$

ryzyko całkowite przyjmuje poniższą postać

$$R^*(\phi) = P_{y,\mathbf{x}}(y \neq \text{sgn}(\mathbf{x}))$$

Niech $P(l|\mathbf{x}) = P(y = l|\mathbf{x})$. Wtedy

$$R^*(\phi) = \int_{\mathbb{X}} \mathbb{I}_{[\phi(\mathbf{x}) \leq 0]} P(1|\mathbf{x}) P(d\mathbf{x}) + \int_{\mathbb{X}} \mathbb{I}_{[\phi(\mathbf{x}) > 0]} P(-1|\mathbf{x}) P(d\mathbf{x})$$

Biorąc skończoną kombinację liniową ± 1 -drzew $\sum_m c_m h_m(\mathbf{x})$, która zbiega do niedodatniej funkcji na zbiorze $\{\mathbf{x} \in \mathbb{X} : P(1|\mathbf{x}) < P(-1|\mathbf{x})\}$ oraz dodatniej na jego dopełnieniu, otrzymujemy, że ryzyko zbiega do ryzyka bayesowskiego

$$\int_{\mathbb{X}} \min(P(1|\mathbf{x}), P(-1|\mathbf{x})) P(d\mathbf{x})$$

Powyższe wyniki gwarantują istnienie liniowej kombinacji ± 1 -drzew zbieżnej do pożądanej funkcji. Postać takiej kombinacji można otrzymać stosując algorytm Gaussa-Southwella opisany przez Breimana w [8] sekcja 3. Wyjściem metody boosting jest funkcja głosowania reprezentowana kombinacją liniową klasyfikatorów z rodziny. Rezultat Breimana opiera się na wykorzystaniu dokładnych wartościach oczekiwanych (dokładnych rozkładów), co implikuje konieczność posiadania informacji dotyczących całej populacji (próby nieskończenie licznej). Gdy próba jest skończona boosting wymaga regularyzacji aby posiadać własność asymptotycznej zbieżności do ryzyka bayesowskiego.

4.5. Modyfikacje algorytmu AdaBoost

Istnieje kilka modyfikacji algorytmu AdaBoost. Modyfikacje te stanowią metody bardziej wyszukane. Algorytm *AdaBoost.OC* (Schapire 1997) jest połączeniem metody AdaBoost i modelu *ECOC*². Takie połączenie daje lepsze wyniki niż sam model ECOC. Rezultaty AdaBoost.OC są porównywalne z wynikami innych modyfikacji, tzn. algorytmem *AdaBoost.M2* (szczególny przypadek algorytmu *AdaBoost.MR*, Schapire i Singer (1999)) oraz algorytmem *AdaBoost.MH* (Schapire i Singer, 1999).

²ECOC - technika *error-correction output coding*, którą zaproponował Dietterich i Bakiri (1991, 1995).

Rozdział 5

Lasy losowe

5.1. Wprowadzenie

Ostatnia rodzina klasyfikatorów, którą będziemy zajmować się w tej pracy, to *lasy losowe*. Doświadczenia eksperymentalne pokazują, że lasy losowe, obok algorytmu boosting, są obecnie najlepszymi klasyfikatorami. Często okazuje się, że lasy losowe dają wyniki lepsze niż te otrzymane metodą boosting. Lasy losowe, inaczej niż algorytmy bagging i boosting, za pojedynczy klasyfikator biorą *drzewo decyzyjne (klasyfikacyjne)*.

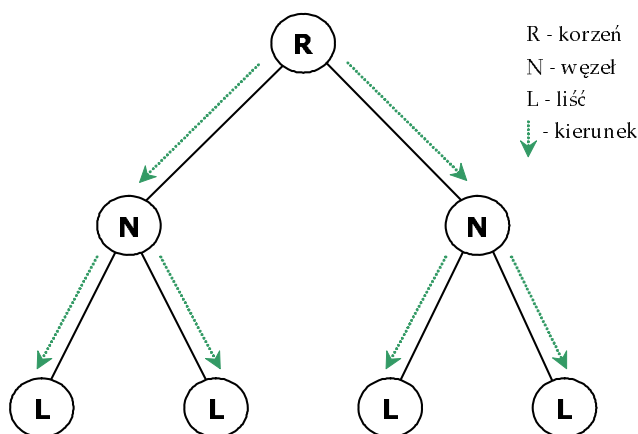
5.2. Drzewa klasyfikacyjne

Drzewa klasyfikacyjne są jedną z najbardziej skutecznych i najpopularniejszych metod klasyfikacji. Podstawową zaletą drzew klasyfikacyjnych jest możliwość reprezentowania dowolnego pojęcia wynikającego z danych uczących. Poza tym, w porównaniu do innych reprezentacji, drzewa charakteryzują się stosunkowo niewielką złożonością obliczeniową i pamięciową. Oprócz zalet związanych z przetwarzaniem maszynowym struktury reprezentującej hipotezy, drzewa są także czytelne dla człowieka, a ewentualne przejście na reprezentację regulową nie stanowi problemu. Aktualnie drzewa klasyfikacyjne wykorzystuje się najczęściej do rozwiązywania problemów z ogromnymi ilościami danych. Poniżej przedstawimy jedynie główne kwestie dotyczące drzew klasyfikacyjnych. Bardziej wyczerpujące opracowanie tematu można znaleźć w [1], [2], [3].

5.2.1. Struktura drzewa

Pojęcie drzewa wywodzi się z teorii grafów. Drzewem nazywamy graf spójny i acykliczny. Interpretację graficzną struktury drzewa klasyfikacyjnego przedstawia rys. 5.1. W drzewie wyróżniamy jeden wierzchołek i nazywamy go *korzeniem*. Każdy liść drzewa nie będący jego korzeniem nazywamy *liściem drzewa klasyfikacyjnego*. Każdy wierzchołek drzewa nie będący liściem drzewa klasyfikacyjnego nazywamy *węzłem drzewa*. Korzeń w drzewie wyznacza kierunek, rozumiany jako kierunek na ścieżce łączącej korzeń z dowolnym wierzchołkiem w drzewie, w szczególności z liściem. Zgodnie z acyklicznością grafu, od korzenia do ustalonego liścia prowadzi tylko jedna droga.

W korzeniu drzewa skupiony jest dostępny zbiór obserwacji. Elementy tego zbioru przesuwane są wzdłuż gałęzi przez węzły. W węzłach podejmowane są decyzje o wyborze gałęzi, wzdłuż której będzie trwać przesuwanie. W ten sposób w każdym węźle dokonywany jest podział



Rysunek 5.1: Struktura drzewa klasyfikacyjnego.

elementów na podgrupy (względem zapisanego w nim *kryterium podziału*). Reguły podziału podpróby docierającej do węzła powinny prowadzić do dobrze określonej maksymalizacji jednorodności ze względu na przynależność do klas podprób w węzłach potomkach danego węzła. Przesuwanie trwa do momentu, gdy napotkamy liść drzewa, który ma etykietę którejś z klas. Rysunek 5.2 przedstawia przykład drzewa klasyfikacyjnego.

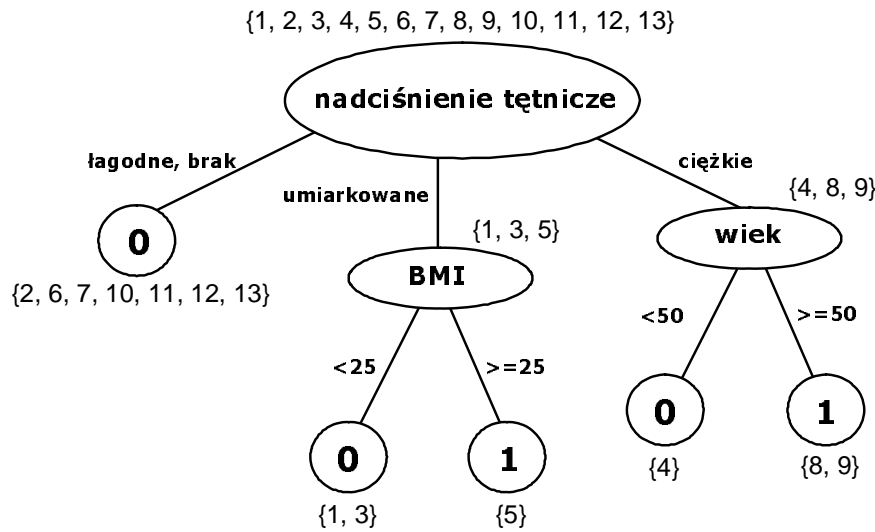
Drzewo klasyfikacyjne jest drzewem, które posiada dodatkową interpretację dla węzłów, gałęzi i liści:

- węzły odpowiadają testom przeprowadzonym na wartościach atrybutów przykładów,
- gałęzie odpowiadają możliwym wynikom tych testów,
- liście odpowiadają etykietom klas rozważanego problemu dyskryminacji,
- drzewo „rośnie” od góry do dołu (od korzenia do liści).

5.2.2. Kryteria podziałów

Wspomnieliśmy już, że podział obserwacji docierających do węzła dokonywany jest względem zapisanego w nim kryterium podziału. Przez regułę podziału rozumiemy funkcję obserwacji $\mathbf{x} \in \mathbb{X}$, przyporządkowaną do węzła, której wartość determinuje przynależność testowanej obserwacji do bezpośredniego potomka rozważanego węzła. Kryteria podziałów nazywamy również *testami*. W praktyce stosuje się najczęściej *testy binarne*, tzn. testy o dwuelementowym zbiorze wyników. Testy binarne generują *drzewa binarne*, w których każdy węzeł ma po dwóch bezpośrednich potomków. W ogólności liczność zbioru możliwych wyników danego testu wyznacza liczbę bezpośrednich potomków węzła, do którego ten test jest przyporządkowany.

Test będąc funkcją obserwacji $\mathbf{x} \in \mathbb{X}$ jest w istocie funkcją wartości atrybutów opisujących rozważane dane. Kryteria podziałów opierają się na testowaniu atrybutów. W praktyce wynik testu uzależniony jest najczęściej od wartości pojedynczego atrybutu. Możliwe jest użycie większej liczby atrybutów w jednym teście, jednak jest to proces kosztowny obliczeniowo i



Rysunek 5.2: Przykład drzewa klasyfikacyjnego.

nawet przy osiągniętym uproszczeniu drzewa okazuje się być nieopłacalnym.

Konstrukcja testów jest wysoce uzależniona od typu testowanego atrybutu. Dla atrybutów nominalnych możemy stosować testy *tożsamościowe*, *równościowe* lub *przynależnościowe*. Atrybuty porządkowe lub ciągłe testujemy testami *nierównościowymi*. Definicje wspomnianych typów testów znajdują się w [2].

5.2.3. Kryteria jakości podziałów

Wybór testu w danym węźle nie powinien być procesem przypadkowym. Zależy nam na uzyskaniu możliwie małej różnorodności klas w otrzymanych częściach (podpróbach), tak aby różnica pomiędzy różnorodnością obserwacji w węźle i różnorodnością klas obserwacji w tych częściach, była możliwie duża. Konieczne jest zatem podanie *miary różnorodności klas* w danym zbiorze obserwacji oraz *miary zmiany różnorodności klasy* po dokonaniu danego podziału. Test maksymalizujący miarę zmiany różnorodności klasy jest testem najlepszym.

Definicja 5.2.1 Każdą funkcję

$$\Phi : G \subseteq [0, 1]^g \rightarrow \mathbb{R}, \quad \text{gdzie dla } (p_1, p_2, \dots, p_g) \in G \text{ zachodzi } \sum_{k=1}^g p_k = 1 \quad (5.1)$$

spełniającą następujące warunki:

1. Φ przyjmuje wartość maksymalną wtedy i tylko wtedy, gdy $p_1 = p_2 = \dots = p_g = \frac{1}{g}$,
2. Φ osiąga minimum tylko w punktach $(1, 0, 0, \dots, 0), (0, 1, 0, \dots, 0), \dots, (0, 0, 0, \dots, 1)$,
3. Φ jest symetryczna ze względu na p_1, p_2, \dots, p_g ,

nazywamy funkcją różnorodności klas.

Zauważmy, że każda funkcja różnorodności klas jest w rzeczywistości funkcją pewnego rozkładu prawdopodobieństwa $\{p_1, p_2, \dots, p_g\}$, w szczególności rozkładu, gdzie p_k jest prawdopodobieństwem klasy k w rozważanym zbiorze obserwacji.

Do najpopularniejszych miar różnorodności klas należą:

1. Proporcja błędnych klasyfikacji:

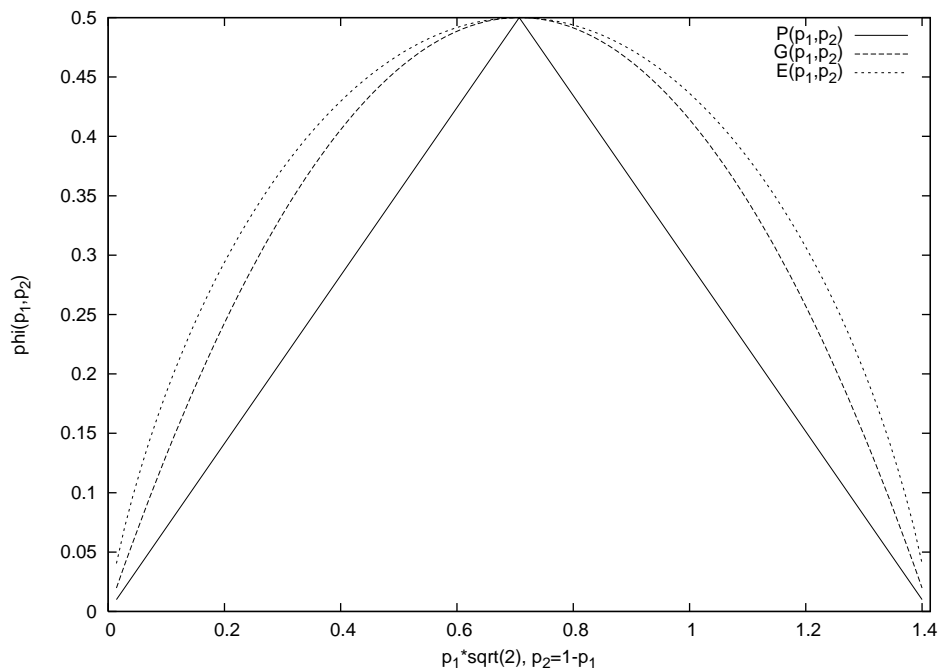
$$P = 1 - \max_k p_k \quad (5.2)$$

2. Indeks Giniego:

$$G = 1 - \sum_{k=1}^g p_k^2 \quad (5.3)$$

3. Entropia:

$$E = - \sum_{k=1}^g p_k \log p_k \quad (5.4)$$



Rysunek 5.3: Proporcja błędnych klasyfikacji, indeks Giniego i entropia

5.2.4. Przycinanie drzew

Drzewa klasyfikacyjne poprzez zdolność reprezentacji dowolnej wynikającej z danych uczących narażone są na nadmierne dopasowanie się do tych danych (tzw. *przeuczenie* lub *przetrenowanie*). Przetrenowanie przejawia się małym błędem na próbce uczącej, ale bardzo dużym błędem rzeczywistym (na próbce testowej). Przycinanie drzewa polega na celowym zwiększeniu błędu klasyfikacji w obszarze danych uczących, w nadziei na uzyskanie mniejszego

błędu rzeczywistego. Podczas procesu przycinania, drzewo wyjściowe zostaje zastąpione swoim poddrzewem. Mówiąc bardziej obrazowo, niektóre poddrzewa drzewa wyjściowego zostają ucięte, a następnie zastąpione liśćmi, którym zostaje przypisana etykieta większościowej kategorii wśród obserwacji związanych z danym poddrzewem. Zamiast przycinać drzewo, możemy poprzez modyfikację kryterium stopu, zapobiec nadmiernemu jego rozrostowi. Taki zabieg nazywa się *przycinaniem podczas wzrostu*. W rzeczywistości jednak znalezienie odpowiedniego kryterium stopu jest trudne i dlatego przycina się drzewo już zbudowane.

5.3. Metoda lasów losowych

Załóżmy, że dany mamy zbiór uczący $\mathcal{L} \in \mathbb{L}$ posiadający n elementów. Na podstawie zbioru \mathcal{L} tworzymy K pseudoprób metodą bootstrap $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_K$ (3.2). Wykorzystując każdą z utworzonych pseudoprób budujemy drzewa klasyfikacyjne, gdzie proces budowy modyfikujemy tak, aby w każdym węźle dokonać wyboru najlepszego podziału na podstawie m wylosowanych atrybutów (pomijając pozostałe). Drzewa budujemy bez przycinania i jeśli to możliwe do otrzymania liści zawierających obserwacje tylko z jednej klasy. W ten sposób otrzymujemy K drzew t_1, t_2, \dots, t_K . W ogólnej sytuacji postać drzewa t_k zależy od pseudopróby \mathcal{L}_k , liczby losowanych atrybutów m oraz od parametru θ_k (np. wagi), gdzie $\{\theta_k\}$ są niezależnymi wektorami losowymi o tym samym rozkładzie. Możemy zapisać $t_k := T(\mathcal{L}_k, m, \theta_k)$.

W myśl definicji 2.2.1 rodzina

$$\mathcal{F} = \left\{ T(\mathcal{L}_k, m, \theta_k) : \mathbb{X} \rightarrow \{1, 2, \dots, g\} \right\}_{k=1,2,\dots,K} \quad (5.5)$$

jest rodziną klasyfikatorów. Wykorzystując regułę głosowania (def. 2.2.3) otrzymujemy klasyfikator $d_{\mathcal{F}}$ generowany rodziną \mathcal{F} . Klasyfikator $d_{\mathcal{F}}$ nazywamy klasyfikatorem otrzymanym metodą lasów losowych (algorytm 5.1).

Algorytm 5.1 Algorytm Forest-RI

```

1  Dane wejściowe:  $\mathcal{L}$ —próba ucząca,  $K$ —liczba iteracji
2  for  $k=1$  to  $K$  do
3      (a) Z próby uczącej  $\mathcal{L}$  wygeneruj pseudopróbkę  $\mathcal{L}_k$ 
4          metodą bootstrap
5      (b) Iteracja dla: wszystkich węzłów  $w$  w drzewie  $t_k$  dopóki
6          nie są spełnione warunki zastopowania budowy drzewa
7          (i) Wybierz  $m$  atrybutów do podziału
8          (ii) Dla każdego z nich wybierz najlepszy podział
9          (iii) Wykonaj podział elementów znajdujących się w węźle
10              $w$  oparty o podział wybrany w (ii)
11  end for
12 Wyjście: Zlicz liczbę głosów  $N_j(\mathbf{x})$ , a następnie oblicz
13      $d_{\mathcal{F}}(\mathbf{x}) = \arg \max_j N_j(\mathbf{x})$ 

```

Modyfikację algorytmu *Forest-RI* przedstawia algorytm 5.2. Metoda *Forest-RC* polega na wybraniu m atrybutów i zsumowaniu ich z wagami wylosowanymi z rozkładu jednostajnego na przedziale $[-1, 1]$. Powtarzając operacje f -krotnie otrzymujemy f nowych atrybutów, z których każdy jest kombinacją liniową m zmiennych.

Algorytm 5.2 Algorytm Forest-RC

```

1 Dane wejściowe:  $\mathcal{L}$ —próbę uczącą,  $K$ —liczbę iteracji
2   for  $k=1$  to  $K$  do
3     (a) Z próby uczącej  $\mathcal{L}$  wygeneruj pseudoprobę  $\mathcal{L}_k$ 
4         metodą bootstrap
5     (b) Iteracja dla: wszystkich węzłów  $w$  w drzewie  $t_k$  dopóki
6         nie są spełnione warunki zastopowania budowy drzewa
7         (i) Wybierz  $m$  atrybutów do podziału
8         (ii) Wygeneruj macierz o wymiarach  $m \times f$  wypełnioną
9             wartościami wylosowanymi z rozkładu jednostajnego na
10            przedziale  $[-1, 1]$ 
11        (iii) Stwórz  $f$  nowych zmiennych i przypisz ich wartości
12            dla każdego elementu  $\mathbf{x}_i$  w węźle  $w$ 
13        (iv) Znajdź najlepsze podziały dla każdej z nowych
14            zmiennych
15        (v) Wykonaj optymalny podział elementów znajdujących się
16            w węźle  $w$ 
17   end for
18 Wyjście: Zlicz liczbę głosów  $N_j(\mathbf{x})$ , a następnie oblicz
19          $d_{\mathcal{F}}(\mathbf{x}) = \arg \max_j N_j(\mathbf{x})$ 

```

Breiman pokazał, że w przypadku tworzenia dużej liczby drzew, nie występuje zjawisko „przeuczenia”, czyli nadmiernego dopasowania się drzew do próby uczącej, ale błąd niepoprawnej klasyfikacji dąży do pewnej wartości. Obrazuje to poniższe twierdzenie.

Twierdzenie 5.3.1 Gdy liczba drzew rośnie, wtedy prawie na pewno błąd $PE^*(\mathcal{F})$ zbiega do

$$P\left(P_{\theta}(T(\mathbf{x}, \theta) = y) < \max_{j \neq y} P_{\theta}(T(\mathbf{x}, \theta) = j)\right).$$

Dowód: Wystarczy pokazać, że istnieje zbiór C o mierze prawdopodobieństwa równej 0, w przestrzeni zdefiniowanej przez ciąg $\theta_1, \theta_2, \dots$ taki, że:

$$\forall \mathbf{x} \notin C \frac{1}{N} \sum_{n=1}^N \mathbb{I}_{[T(\theta_n, \mathbf{x})=j]} \longrightarrow P_{\theta}(T(\theta, \mathbf{x}) = j).$$

Dla ustalonej próby uczącej oraz ustalonego θ , zbiór wszystkich \mathbf{x} takich, że $T(\theta, \mathbf{x}) = j$ jest sumą kostek wielowymiarowych. Dla każdego drzewa $T(\theta, \mathbf{x})$ istnieje skończona liczba K takich sum, które oznaczono S_1, S_2, \dots, S_K . Niech $\varphi(\theta_n) = k$ jeżeli $\{\mathbf{x} : T(\theta, \mathbf{x}) = j\} = S_k$. Przez N_k będziemy oznaczać liczbę przypadków kiedy $\varphi(\theta_n) = k$ w pierwszych N próbach. Możemy wtedy zapisać, że:

$$\frac{1}{N} \sum_{n=1}^N \mathbb{I}_{[T(\theta_n, \mathbf{x})=j]} = \frac{1}{N} \sum_k N_k \mathbb{I}_{[\mathbf{x} \in S_k]}$$

Z prawa wielkich liczb wynika, że

$$N_k = \sum_{n=1}^N \mathbb{I}_{[\varphi(\theta_n)=k]} \longrightarrow P(\varphi(\theta) = k)$$

Po wzięciu sumy po wszystkich zbiorach, dla których ta zależność nie jest prawdziwa dla pewnych wartości k , dostajemy zbiór C o zerowym prawdopodobieństwie. Dla obserwacji z tego zbioru zachodzi więc

$$\frac{1}{N} \sum_{n=1}^N \mathbb{I}_{[T(\theta_n, \mathbf{x})=j]} \longrightarrow \frac{1}{N} \sum_k P(\varphi(\theta) = k) \mathbb{I}_{[\mathbf{x} \in S_k]} = P_\theta(T(\theta, \mathbf{x}) = j)$$

■

5.4. Moc i korelacja

Lasy losowe opierają się bardzo mocno na idei łączenia w jedną rodzinę możliwie dobrych klasyfikatorów, ale jednocześnie możliwie słabo od siebie zależnych. Breiman wykazał, że prawdopodobieństwo popełnienia przez las losowy błędu klasyfikacyjnego rośnie wraz ze wzrostem odpowiednio zdefiniowanego współczynnika korelacji między drzewami i maleje wraz ze wzrostem tzw. siły pojedynczych drzew (jakości klasyfikacji pojedynczego drzewa).

Dla lasów losowych można zdefiniować górną granicę błędu niepoprawnej klasyfikacji w zależności od dwóch parametrów określających dokładność klasyfikacji.

Definicja 5.4.1 *Marginesem klasyfikacji* obserwacji $\mathbf{x} \in \mathbb{X}$ z klasy $y \in \{1, 2, \dots, g\}$ dla lasów losowych \mathcal{F} nazywamy funkcję

$$\text{mg}_{\mathcal{F}}(\mathbf{x}, y) := P_\theta(T(\mathbf{x}, \theta) = y) - \max_{j \neq y} P_\theta(T(\mathbf{x}, \theta) = j). \quad (5.6)$$

Definicja 5.4.2 Moc zbioru klasyfikatorów $\{T(\mathbf{x}, \theta)\}$ określamy jako

$$s := \mathbb{E}[\text{mg}_{\mathcal{F}}(\mathbf{x}, y)].$$

Lemat 5.4.1 Jeżeli θ_1 i θ_2 mają ten sam rozkład i są niezależne, to dla dowolnej funkcji f prawdziwa jest równość:

$$[\mathbb{E}_{\theta_1} f(\theta_1)]^2 = \mathbb{E}_{\theta_1, \theta_2} f(\theta_1) f(\theta_2)$$

Dowód: Jeżeli θ_1, θ_2 są iid, to $\mathbb{E}_{\theta_1} f(\theta_1) = \mathbb{E}_{\theta_2} f(\theta_2)$, czyli

$$[\mathbb{E}_{\theta_1} f(\theta_1)]^2 = \mathbb{E}_{\theta_1} f(\theta_1) \mathbb{E}_{\theta_1} f(\theta_1) = \mathbb{E}_{\theta_1} f(\theta_1) \mathbb{E}_{\theta_2} f(\theta_2) = \mathbb{E}_{\theta_1, \theta_2} f(\theta_1) f(\theta_2)$$

■

Twierdzenie 5.4.1 Górna granica dla błędu $PE^*(\mathcal{F})$ jest dana przez wartość

$$PE^*(\mathcal{F}) \leq \bar{\rho} \frac{1 - s^2}{s^2} \quad (5.7)$$

gdzie $\bar{\rho}$ reprezentuje korelację między drzewami lasu losowego oraz $s \geq 0$, co oznacza, że błąd pojedynczego klasyfikatora jest mniejszy niż błąd losowego zgadywania klasy.

Dowód: Jeśli $s \geq 0$, to możemy skorzystać z nierówności Czybyszewa i otrzymamy

$$PE^*(\mathcal{F}) \leq \frac{\text{Var}(\text{mg}_{\mathcal{F}}(\mathbf{x}, y))}{s^2}$$

Jeżeli przyjmiemy, że

$$j^*(\mathbf{x}, y) = \arg \max_{j \neq y} P_\theta(T(\mathbf{x}, \theta) = j)$$

to możemy zdefiniować *czystą funkcję marginesu klasyfikacji* jako

$$\text{cmg}_{\mathcal{F}}(\theta, \mathbf{x}, y) = \mathbb{I}_{[T(\mathbf{x}, \theta)=y]} - \mathbb{I}_{[T(\mathbf{x}, \theta)=j^*(\mathbf{x}, y)]}$$

Zapisując funkcję marginesu klasyfikacji w następujący sposób

$$\text{mg}_{\mathcal{F}}(\mathbf{x}, y) = P_\theta(T(\mathbf{x}, \theta) = y) - P_\theta(T(\mathbf{x}, \theta) = j^*(\mathbf{x}, y)) = \mathbb{E}_\theta(\mathbb{I}_{[T(\mathbf{x}, \theta)=y]} - \mathbb{I}_{[T(\mathbf{x}, \theta)=j^*(\mathbf{x}, y)]})$$

to widać, że $\text{mg}_{\mathcal{F}}(\mathbf{x}, y)$ jest wartością oczekiwaną czystej funkcji marginesu klasyfikacji $\text{cmg}_{\mathcal{F}}(\theta, \mathbf{x}, y)$, czyli

$$\text{mg}_{\mathcal{F}}(\mathbf{x}, y) = \mathbb{E}_\theta(\text{cmg}_{\mathcal{F}}(\theta, \mathbf{x}, y))$$

Na podstawie lematu 5.4.1

$$[\text{mg}_{\mathcal{F}}(\mathbf{x}, y)]^2 = \mathbb{E}_{\theta_1, \theta_2}(\text{cmg}_{\mathcal{F}}(\theta_1, \mathbf{x}, y) \text{cmg}_{\mathcal{F}}(\theta_2, \mathbf{x}, y))$$

Korzystając ze wzoru na wariancję otrzymujemy

$$\text{Var}(\text{mg}_{\mathcal{F}}) = \mathbb{E}_{\theta_1, \theta_2}(\text{Cov}(\text{cmg}_{\mathcal{F}}(\theta_1, \mathbf{x}, y), \text{cmg}_{\mathcal{F}}(\theta_2, \mathbf{x}, y))) = \mathbb{E}_{\theta_1, \theta_2}(\rho(\theta_1, \theta_2) \sigma(\theta_1) \sigma(\theta_2))$$

gdzie $\rho(\theta_1, \theta_2)$ jest korelacją pomiędzy $\text{cmg}_{\mathcal{F}}(\theta_1, \mathbf{x}, y)$ i $\text{cmg}_{\mathcal{F}}(\theta_2, \mathbf{x}, y)$ dla ustalonych θ_1 i θ_2 , a $\sigma(\theta)$ jest odchyleniem standardowym $\text{cmg}_{\mathcal{F}}(\theta, \mathbf{x}, y)$ przy ustalonym θ . Jeżeli przez $\bar{\rho}$ będziemy rozumieli wartość oczekiwaną $\rho(\theta_1, \theta_2)$ równą

$$\bar{\rho} = \mathbb{E}_{\theta_1, \theta_2}(\rho(\theta_1, \theta_2) \sigma(\theta_1) \sigma(\theta_2)) / \mathbb{E}_{\theta_1, \theta_2}(\sigma(\theta_1) \sigma(\theta_2))$$

to

$$\text{Var}(\text{mg}_{\mathcal{F}}) = \bar{\rho} \left(\mathbb{E}_\theta \sigma(\theta) \right)^2 \leq \bar{\rho} \mathbb{E}_\theta \text{Var}(\theta) \leq \mathbb{E}_\theta \left(\mathbb{E} \text{cmg}_{\mathcal{F}}(\theta, \mathbf{x}, y) \right)^2 - s^2 \leq 1 - s^2$$

Podsumowując otrzymujemy górne ograniczenie $PE^*(\mathcal{F})$ równe $\bar{\rho} \frac{1-s^2}{s^2}$. ■

Z powyższego twierdzenia wynika, że błąd klasyfikacji można oszacować przy użyciu dwóch wartości: mocy poszczególnych klasyfikatorów lasu losowego oraz korelacji między nimi. Zdefiniujmy miarę c/s^2 będącą ilorazem korelacji i kwadratu mocy, czyli

$$c/s^2 = \frac{\bar{\rho}}{s^2}$$

Okazuje się zatem, że im mniejsza wartość tej miary, tym lepsze wyniki klasyfikacji można otrzymać przy zastosowaniu lasu losowego.

Rozdział 6

Analiza danych

6.1. Wprowadzenie

Analiza danych została przeprowadzona przy użyciu funkcji dostępnych w pakiecie R, [12]. Wykorzystano moduły:

- **adabag** - implementacja metody bagging oraz algorytmu AdaBoost.M1, [14]
- **randomForest** - implementacja lasów losowych, [15]
- **rpart** - implementacja drzew decyzyjnych, [16]
- **mlbench** - zbiory danych, [17]

Celem analizy była prezentacja własności opisanych metod łączenia klasyfikatorów. Analizą objęto dane z modułu **mlbench**:

- **BreastCancer** - dane dotyczące zachorowań na raka piersi, zebrane przez Dr. William H. Wolberg, University of Wisconsin Hospital, Madison, Wisconsin, USA (699 obserwacji, 11 atrybutów, 9 ciągłych, 1 nominalny, 1 atrybut grupujący).
- **Ionosphere** - dane radarowe dotyczące obserwacji wolnych elektronów w jonosferze, zebrane przez system Goose Bay, Labrador (351 obserwacji, 35 atrybutów, 32 ciągłe, 2 nominalne, 1 atrybut grupujący).
- **PimaIndiansDiabetes** - dane dotyczące zachorowań na cukrzycę wśród Indian Pima (768 obserwacji, 9 atrybutów, 7 ciągłych, 1 nominalny, 1 atrybut grupujący).
- **Satellite** - obserwacje z satelity Landsat (Multi-Spectral Scanner Image Data) punktów w 3×3 sąsiedztwie, dostępna klasyfikacja punktów centralnych, zebrane przez NASA (6435 obserwacji, 36 atrybutów, 1 atrybut grupujący).
- **Shuttle** - dane udostępnił Jason Catlett z uniwersytetu w Sydney (58000 obserwacji, 9 atrybutów ciągłych, 1 atrybut grupujący).
- **Sonar** - dane sonarowe dotyczące odróżnienia min od skał, udostępnił Terry Sejnowski, Salk Institute and University of California, San Deigo (208 obserwacji, 61 atrybutów, 60 ciągłych, 1 atrybut grupujący).
- **Vehicle** - charakterystyki typów pojazdów, udostępnione przez Drs.Pete Mowforth and Barry Shepherd, Turing Institute, Glasgow, Scotland (846 obserwacji, 19 atrybutów, 18 ciągłych, 1 atrybut grupujący).

- **Vowel** - rozpoznawanie samogłosek, udostępnił Tony Robinson (990 obserwacji, 10 atrybutów, 8 ciągłych, 1 nominalny, 1 atrybut grupujący).
- **HippoSeqFeature** - zbiór danych z analizy obszarów regulatorowych i ekspresji genów w trakcie rozwoju hipokampa (część mózgu) u myszy, udostępnił Dr Michał Dąbrowski (2021 obserwacji, 149 atrybutów zero-jedynkowych, w tym jeden atrybut grupujący).

Dodatkowo wygenerowano dwuwymiarowy zbiór obserwacji (współrzędne na płaszczyźnie XY) z podziałem na dwie klasy. Całość obliczeń realizowały dwa komputery PC pracując nieustannie przez około 168 godzin.

6.2. Granica podziału

Przypomnijmy, że w rozdziale 2 pisaliśmy o możliwości wystąpienia braku w przestrzeni hipotez dokładnego rozwiązania problemu dyskryminacji. W sekcji 2.3 twierdziliśmy, że rodziny klasyfikatorów mogą wygenerować rozwiązanie znacznie bliższe rozwiązaniu dokładnemu. Na rysunku 6.1 prezentujemy wyniki eksperymentu, który bada zmianę błędu i kształtu klas rozwiązania. Tabela 6.1 zawiera zestawienie błędów dla rozważanych metod w poszczególnych iteracjach.

Błąd pojedynczego drzewa: 0,0586

Liczba iteracji	Bagging	AdaBoost.M1	Las losowy
1	0,0486	0,0625	0,0369
5	0,0547	0,0217	0,0308
99	0,0521	0,0200	0,0213

Tabela 6.1: Bagging, AdaBoost.M1, Las losowy - zamiana błędu dla eksperymentu przedstawionego na rysunku 6.1.

Rysunek 6.1 a) przedstawia dwuwymiarową przestrzeń obserwacji (współrzędne XY punktów) oraz podział na dwie klasy: klasa 1 - czerwone „koło”, klasa 2 - punkty niebieskie. Rysunek oznaczony literą b) prezentuje zbiór uczący wylosowany jednostajnie z przestrzeni obserwacji a). Wynik klasyfikacji pojedynczym drzewem opisuje część c) rysunku 6.1. Dobrze widoczne jest „kanciaste” dopasowanie do kształtu koła klasy 1. Kolejne wykresy przedstawiają kształt klasy 1 po zastosowaniu agregacji metodą bagging (d, g, j), boosting (e, h, k) oraz metodą lasów losowych (f, i, l). Nietrudno zauważyć, że agregacja zbliża kształt klasy 1 do wyjściowego koła. Dobrze widoczne są również różnice w dopasowaniu pomiędzy wymienionymi metodami agregacji. Metodą bagging otrzymaliśmy wynik zdecydowanie gorszy niż pozostałymi dwiema metodami. Znaczne zwiększenie liczby iteracji (z 5 do 99) również nie wpływa na znaczną poprawę kształtu. Inną sytuację obserwujemy dla metody boosting i lasu losowego. Wykorzystując algorytm AdaBoost.M1 już po 5 iteracjach otrzymaliśmy kształt dobrze przybliżający koło klasy 1. Tym razem zwiększenie liczby iteracji zdecydowanie wpływa na poprawę jakości dopasowania. Podobny efekt daje las losowy, gdzie już pierwsza iteracja odsłania dość gładką granicę podziału. Najmniejszy błąd osiągnęliśmy metodą boosting.

6.3. Błąd klasyfikacji

Najbardziej istotną zaletą rodzin klasyfikatorów jest poprawa jakości klasyfikacji. Przez poprawę jakości klasyfikacji rozumiemy osiągnięcie mniejszego błędu klasyfikacji niż ten otrzy-

many przy wykorzystaniu pojedynczego klasyfikatora. Błąd klasyfikacji szacujemy metodą próby testowej. Analizę przeprowadziliśmy w poniższych krokach

1. Podział zbioru danych na próbę uczącą oraz próbę testową.
2. Wysterowanie klasyfikatora pojedynczego (drzewo regresyjne) przy wykorzystaniu parametrów ograniczających wysokość drzewa oraz minimalną liczbę obserwacji w węźle. Utworzenie klasyfikatora pojedynczego na podstawie próby uczącej, wyznaczenie błędu klasyfikacji na podstawie próby testowej.
3. Utworzenie klasyfikatorów metodami bagging, boosting oraz lasów losowych na podstawie próby uczącej dla iteracji od 1 do 100, oszacowanie błędu klasyfikacji na podstawie próby testowej, 10-krotne powtórzenie procedury.
4. Wyznaczenie średniego błędu klasyfikacji dla każdej z metod w podziale na iteracje od 1 do 100
5. Wyznaczenie wariancji błędu klasyfikacji dla każdej z metod w podziale na iteracje od 1 do 100
6. Graficzna prezentacja oraz analiza wyników.

6.3.1. Analiza danych BreastCancer

Graficzna prezentacja wykonanych kroków 1 – 6 dla danych **BreastCancer** znajduje się na rysunkach 6.2 oraz 6.3. Niebieskie chmury punktów na rysunku 6.2 a), b), c) przedstawiają wyniki 10 powtórzeń iteracji od 1 do 100 dla metod a) bagging, b) boosting, c) lasu losowego. Czarna przerywana linia to błąd klasyfikacji pojedynczego drzewa. Czerwona kreska wskazuje wartość średnią błędu klasyfikacji w 10 powtórzeniach. Z łatwością zauważamy, że już po kilku iteracjach uśredniony błąd klasyfikatorów zagregowanych jest mniejszy niż błąd pojedynczego drzewa. Dobrze widoczna jest stabilizacja średniego błędu klasyfikacji wraz ze wzrostem liczby iteracji. Warto również zwrócić uwagę, że dla iteracji od 1 do około 30 część chmury punktów dla metody bagging koncentruje się wokół wartości odpowiadającej błędowi pojedynczego drzewa. Taka sytuacja sugeruje, że bagging jest bardziej podatny na generowanie klasyfikatorów nie lepszych niż reguły podstawowe. Porównanie metody bagging, boosting oraz lasów losowych przedstawia rysunek 6.2 d). Najlepsze wyniki otrzymaliśmy metodą lasów losowych. Błąd dla metod bagging i boosting jest na podobnym poziomie.

Wariancje błędów klasyfikacji dla każdej z wymienionych metod przedstawione są na rysunku 6.3 a) bagging, b) boosting, c) las losowy, d) trzy metody na jednym wykresie. Możemy jednoznacznie stwierdzić, że dla danych **BreastCancer** agregacja klasyfikatorów zmniejsza wariancję błędu klasyfikacji wraz ze wzrostem liczby iteracji.

6.3.2. Analiza danych Vowel

Graficzna prezentacja wykonanych kroków 1 – 6 dla danych **Vowel** znajduje się na rysunkach 6.4 oraz 6.5. Niebieskie chmury punktów na rysunku 6.4 a), b), c) przedstawiają wyniki 10 powtórzeń iteracji od 1 do 100 dla metod a) bagging, b) boosting, c) lasu losowego. Czarna przerywana linia to błąd klasyfikacji pojedynczego drzewa. Czerwona kreska wskazuje wartość średnią błędu klasyfikacji w 10 powtórzeniach. Ponownie z łatwością zauważamy, że już po kilku iteracjach uśredniony błąd klasyfikatorów zagregowanych jest mniejszy niż błąd pojedynczego drzewa. Tu również dobrze widoczna jest stabilizacja średniego błędu klasyfikacji

wraz ze wzrostem liczby iteracji. Tym razem efekt koncentracji części chmury punktów dla metody bagging wokół wartości odpowiadającej błędowi pojedynczego drzewa nie jest widoczny. Porównanie metody bagging, boosting oraz lasów losowych przedstawia rysunek 6.4 d). Najlepsze wyniki otrzymaliśmy metodą boosting. Błąd otrzymany metodą lasów losowych jest średnio mniejszy niż błąd dla metody bagging.

Wariancje błędów klasyfikacji dla każdej z wymienionych metod przedstawione są na rysunku 6.5 a) bagging, b) boosting, c) las losowy, d) trzy metody na jednym wykresie. Tym razem jedynie nieśmiało możemy stwierdzić, że dla danych Vowel agregacja klasyfikatorów zmniejsza wariancję błędu klasyfikacji wraz ze wzrostem liczby iteracji.

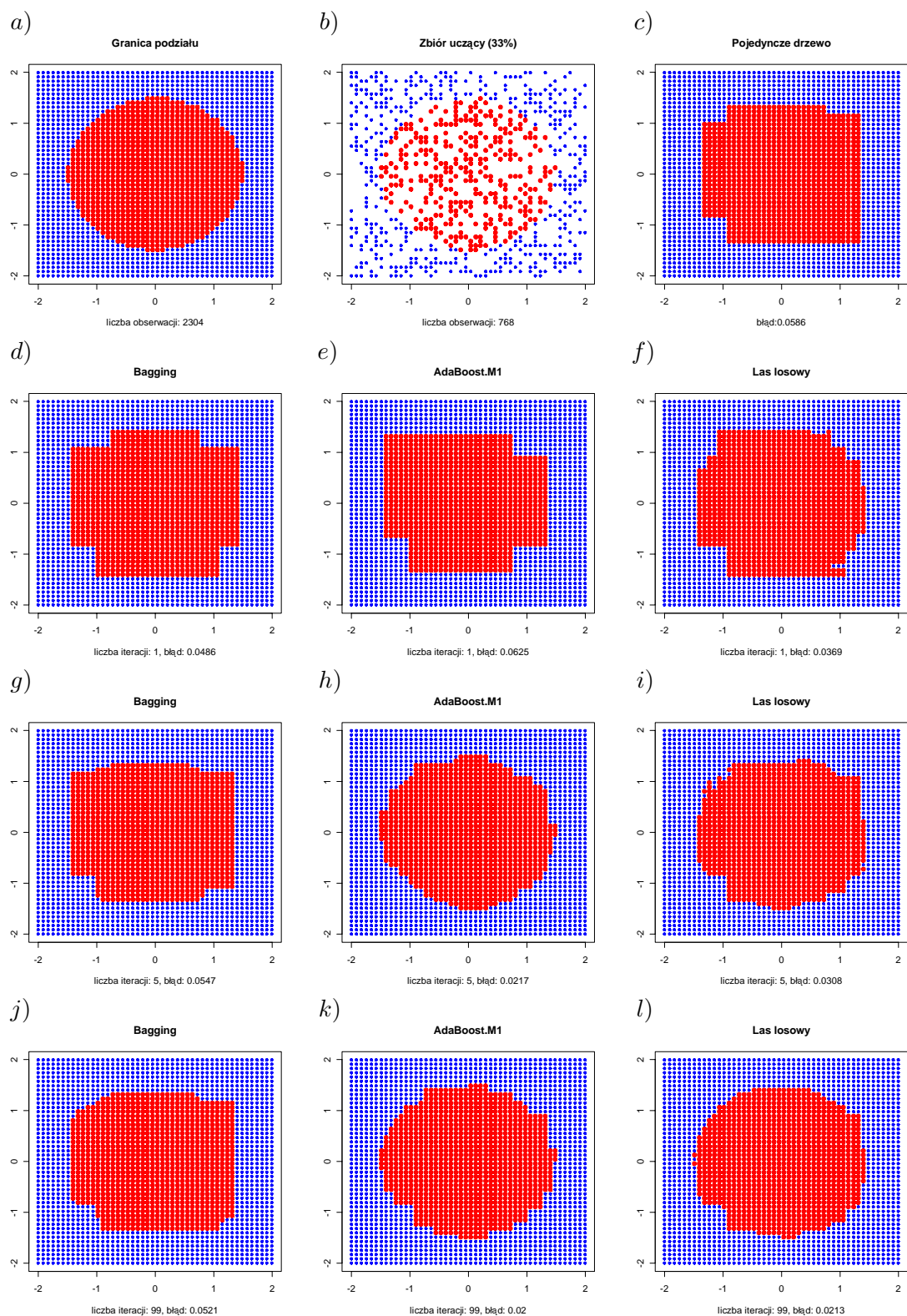
6.3.3. Analiza pozostałych danych

Z powodu czasochłonności obliczeń (długi czas przetwarzania w pakiecie R) analizę pozostałych danych ograniczyliśmy do jednego przebiegu iteracji od 1 do 100. Rysunki 6.6 oraz 6.7 przedstawiają wykresy porównawcze dla metod bagging, boosting, lasu losowego. Wnioski:

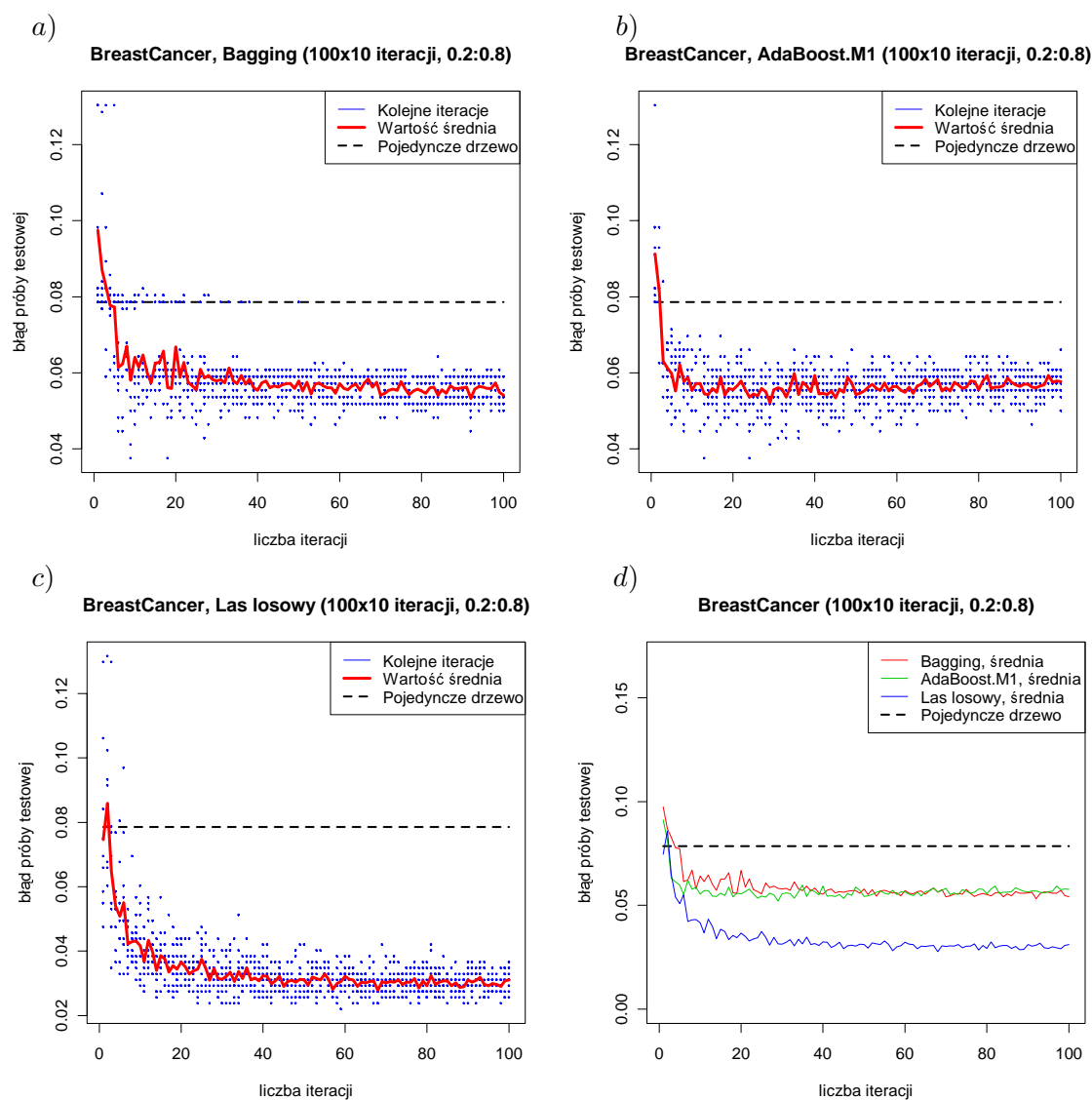
1. **Vehicle**, rys. 6.6 a) - poprawa jakości klasyfikacji, najlepszy wynik dla lasu losowego, największy błąd otrzymany metodą bagging.
2. **Vowel**, rys. 6.6 b) - poprawa jakości klasyfikacji, najlepszy wynik dla lasu losowego, bagging i boosting generują klasyfikatory o podobnej jakości.
3. **BreastCancer**, rys. 6.6 c) - poprawa jakości klasyfikacji dla lasu losowego i boostingu, brak poprawy dla metody bagging, najlepszy wynik dla lasu losowego.
4. **Ionosphere**, rys. 6.6 d) - widoczna poprawa jakości klasyfikacji jedynie dla lasu losowego, najlepszy wynik dla lasu losowego, bagging i boosting generują klasyfikatory o podobnej jakości.
5. **Sonar**, rys. 6.6 e) - poprawa jakości klasyfikacji, duża wariancja, najlepszy wynik dla metody boosting.
6. **PimaIndiansDiabetes**, rys. 6.6 f) - brak poprawy jakości klasyfikacji, wyniki porównywalne dla każdej z metod.
7. **Satellite**, rys. 6.7 a) - poprawa jakości klasyfikacji, najlepszy wynik dla lasu losowego, największy błąd otrzymany metodą bagging.
8. **HippoSeqFeature**, rys. 6.7 b) - niewielka poprawa jakości klasyfikacji metodami bagging i boosting, las losowy daje wyniki gorsze od pojedynczego drzewa, brak poprawy wyników wraz ze wzrostem liczby iteracji (liczby klasyfikatorów). Ciekawą obserwacją jest to, że w przypadku danych ledwie lepszych od losowych (w danych HippoSeqFeature pojedynczy błąd utrzymuje się na poziomie 0.44) bagging i boosting „trzymają” się błędu pojedynczego drzewa natomiast las losowy błąd powiększa. Powyższe można prawdopodobnie tłumaczyć tym, że las losowy sam wprowadza dużo losowości podczas wyboru atrybutów do podziału i tym samym może wypaść gorzej niż drzewo oparte na wszystkich atrybutach.

6.4. Podsumowanie

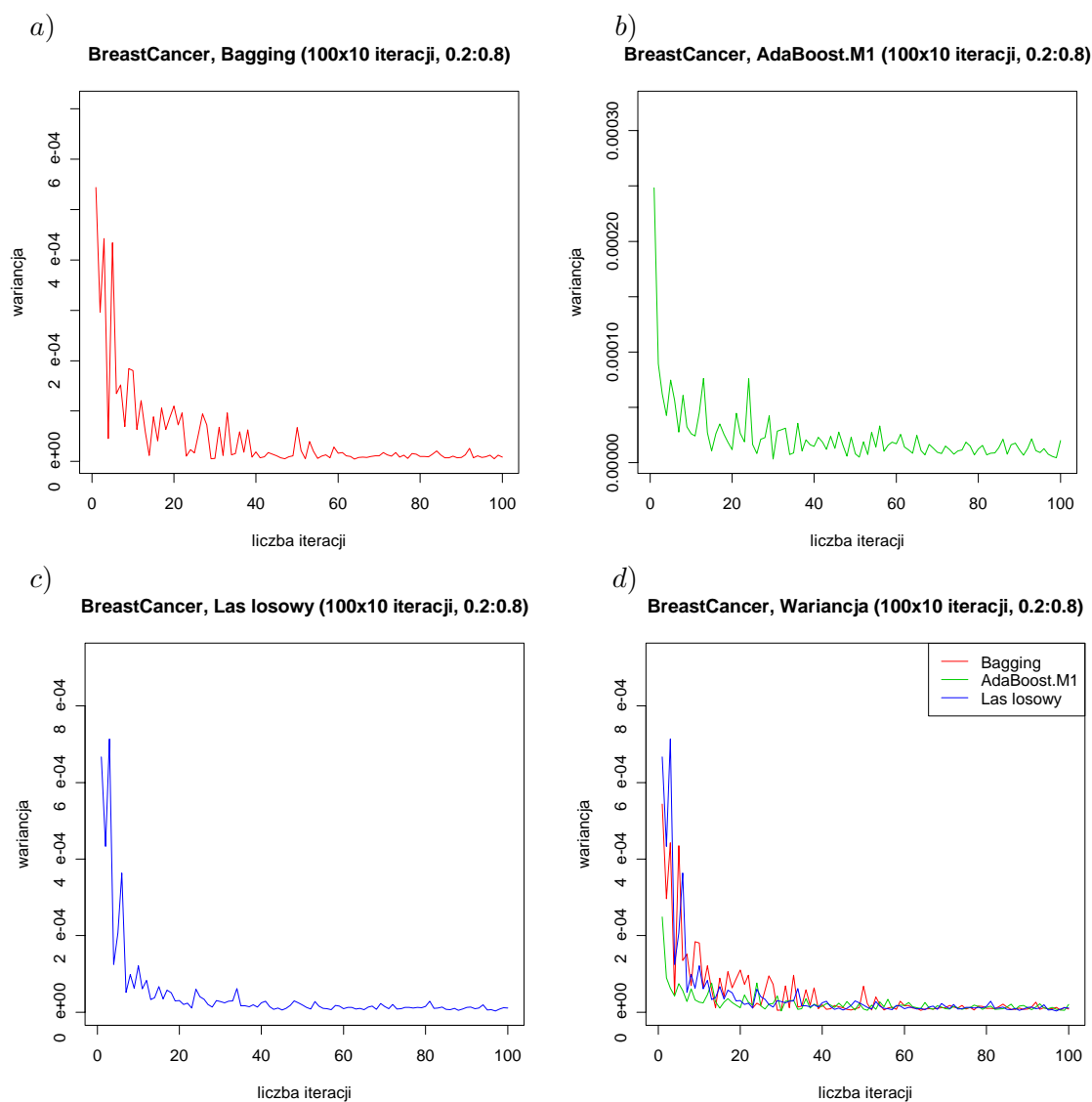
Wyniki analizy danych potwierdzają zasadność konstrukcji rodzin klasyfikatorów. Poprawa jakości klasyfikacji wraz z redukcją wariancji stanowi cenną wartość dodaną nie tylko w przypadkach, gdy dysponujemy jedynie słabymi klasyfikatorami. W większości analizowanych sytuacji las losowy okazał się być najbardziej odpowiednią metodą. Czasami najlepsze wyniki generował algorytm boosting. Natomiast w analizach najgorzej wypadł bagging. W analizach natknęliśmy się również na przypadki gdzie nie dochodziło do poprawy jakości klasyfikacji. Należy zwrócić uwagę, że sposób działania danej metody zależy w dużym stopniu od parametrów eksperymentu (typ problemu, próba ucząca, klasyfikator podstawowy).



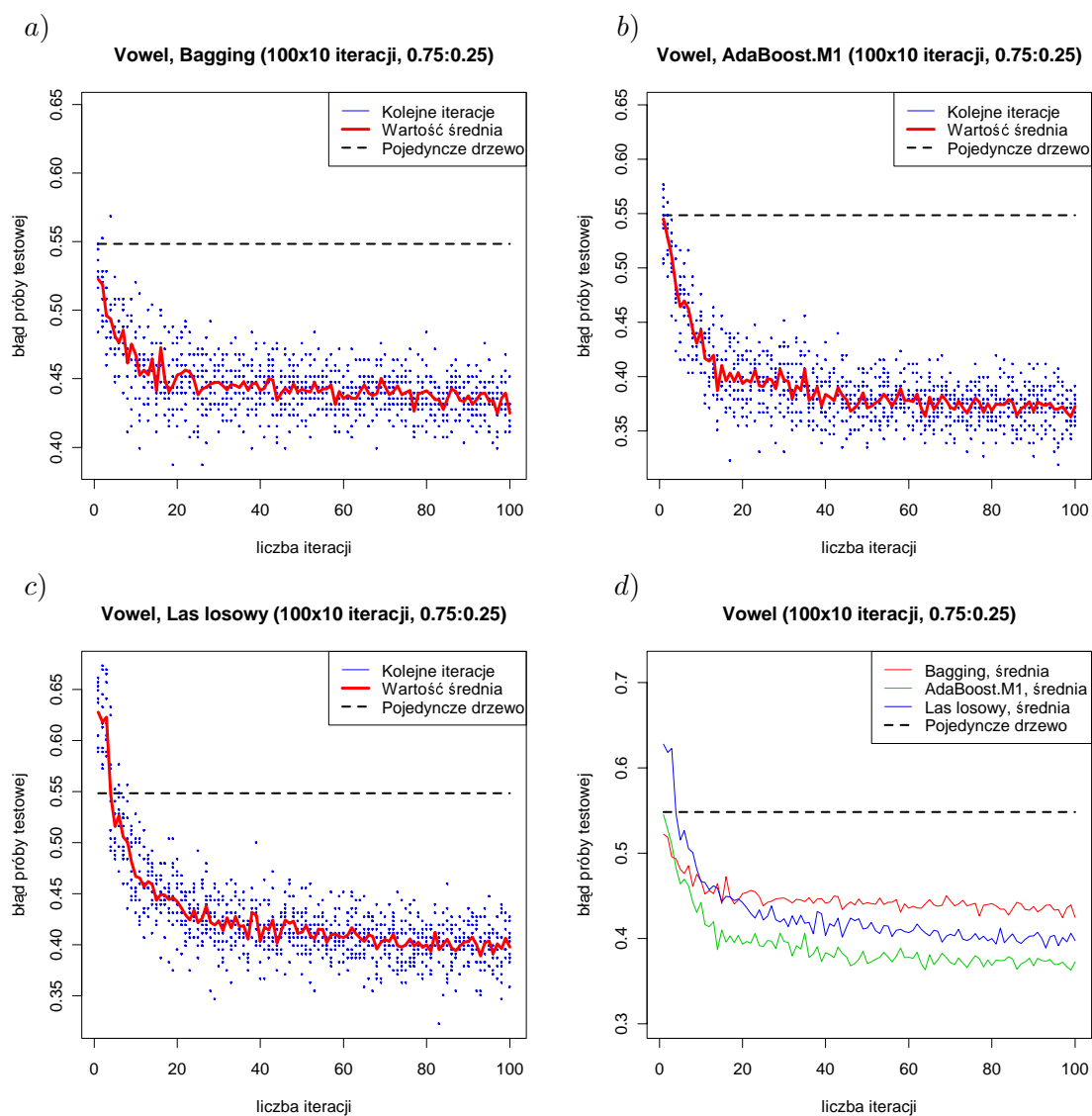
Rysunek 6.1: Efekt przybliżenia do granicy podziału na przykładzie rodzin bagging, boosting (z drzewem decyzyjnym jako klasyfikatorem podstawowym) oraz lasu losowego.



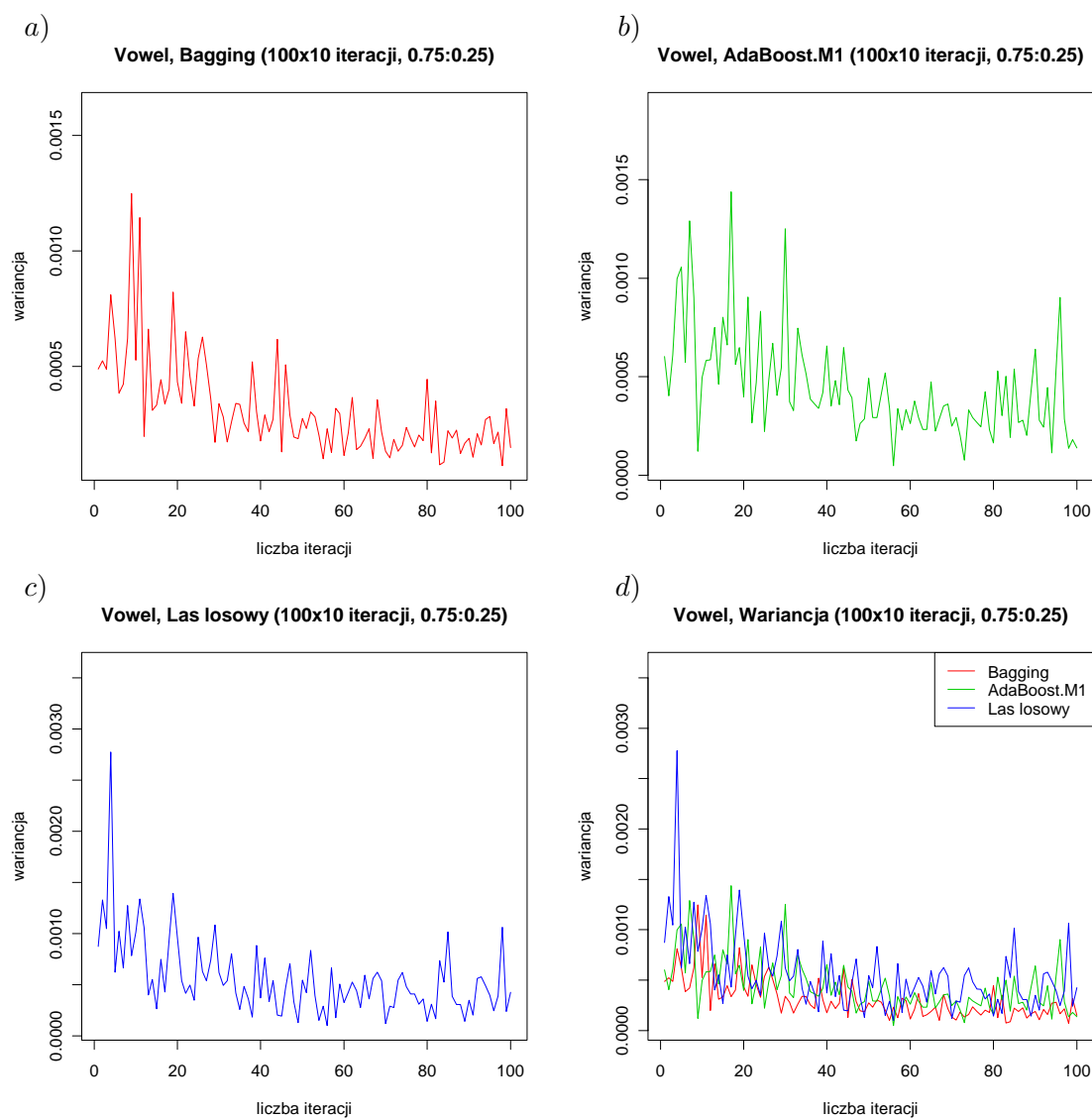
Rysunek 6.2: Średni błąd klasyfikacji dla rodzin bagging, boosting oraz lasu losowego na przykładzie danych BreastCancer (10 powtórzeń od 1 do 100 iteracji): a) bagging, b) boosting, c) las losowy, d) bagging, boosting oraz las losowy.



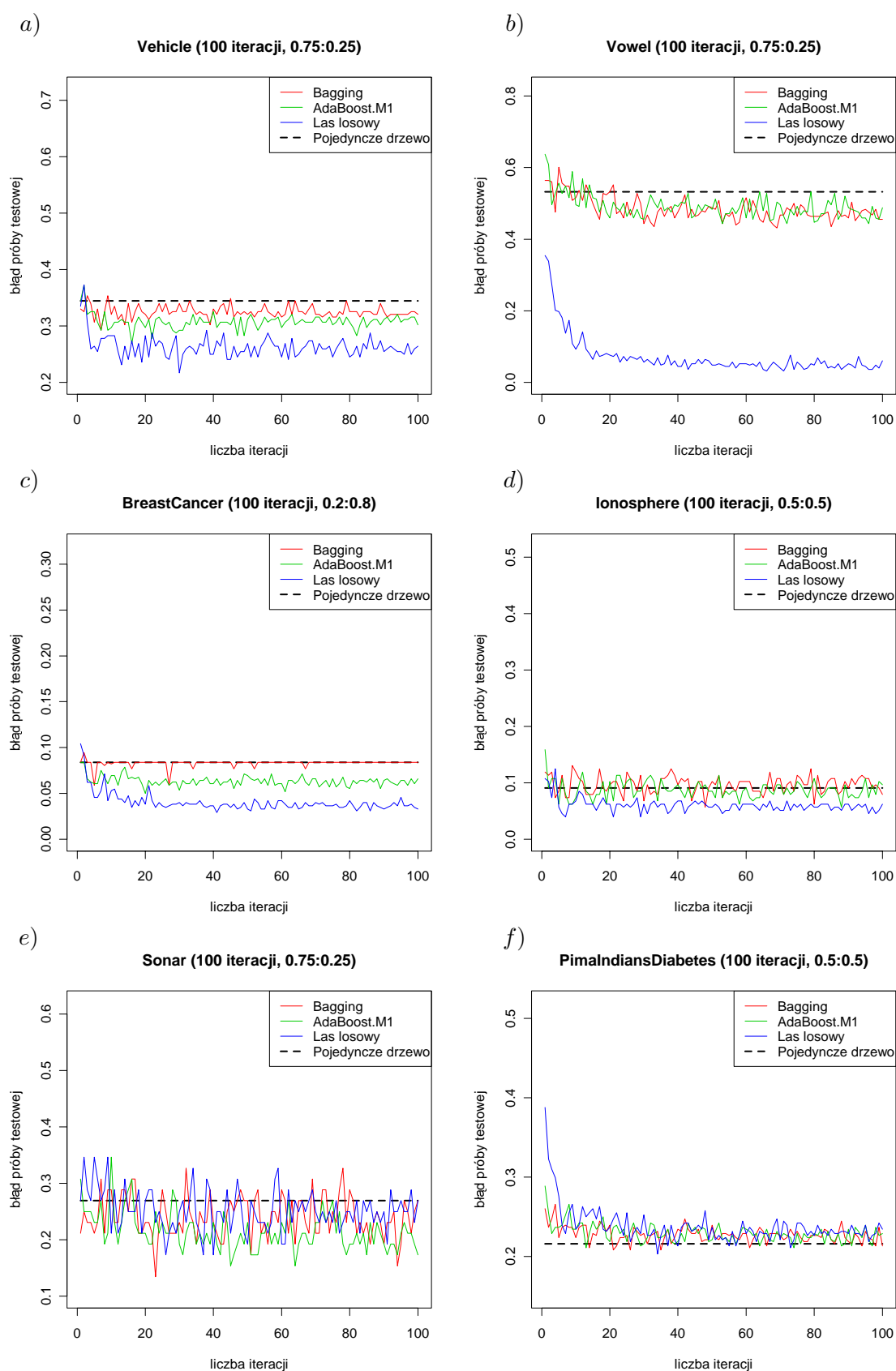
Rysunek 6.3: Wariancja błędu klasyfikacji dla rodzin bagging, boosting oraz lasu losowego na przykładzie danych BreastCancer (10 powtórzeń od 1 do 100 iteracji): a) bagging, b) boosting, c) las losowy, d) bagging, boosting oraz las losowy.



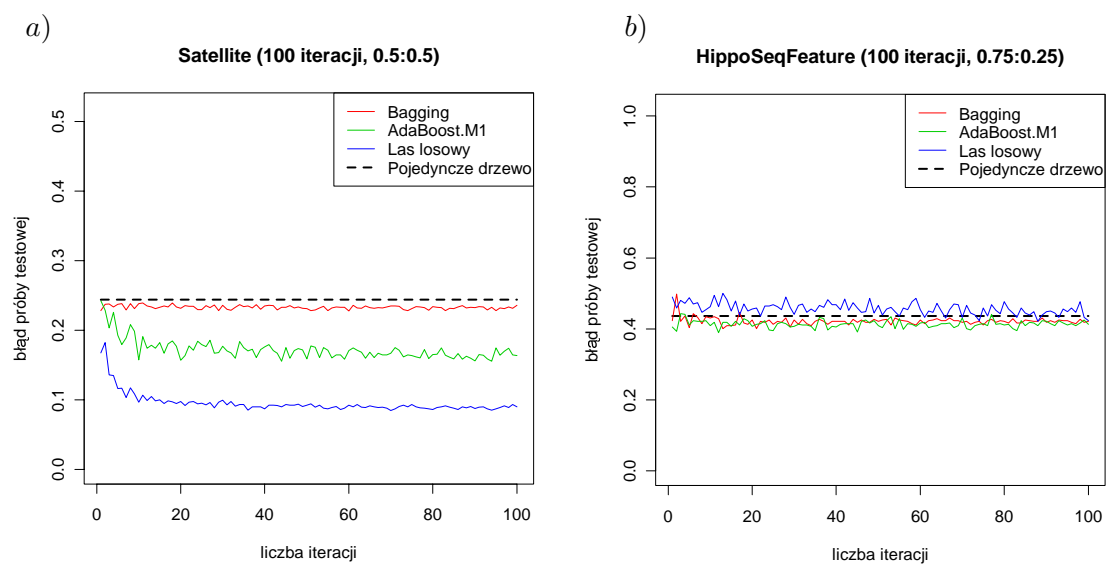
Rysunek 6.4: Średni błąd klasyfikacji dla rodzin bagging, boosting oraz lasu losowego na przykładzie danych Vowel (10 powtórzeń od 1 do 100 iteracji): a) bagging, b) boosting, c) las losowy, d) bagging, boosting oraz las losowy.



Rysunek 6.5: Wariancja błędu klasyfikacji dla rodzin bagging, boosting oraz lasu losowego na przykładzie danych Vowel (10 powtórzeń od 1 do 100 iteracji): a) bagging, b) boosting, c) las losowy, d) bagging, boosting oraz las losowy.



Rysunek 6.6: Porównanie błędu klasyfikacji dla rodzin bagging, boosting oraz lasu losowego na przykładzie danych: a) Vehicle, b) Vowel, c) BreastCancer, d) Ionosphere, e) Sonar, f) PimaIndiansDiabetes.



Rysunek 6.7: Porównanie błędu klasyfikacji dla rodzin bagging, boosting oraz lasu losowego na przykładzie danych: a) Satellite, b) HippoSeqFeature.

Bibliografia

- [1] Jacek Koronacki, Jan Ówik: *Statystyczne systemy uczące się*, Wydawnictwa Naukowo-Techniczne, Warszawa 2005
- [2] Paweł Cichosz: *Systemy uczące się*, Wydawnictwa Naukowo-Techniczne, Warszawa 2000
- [3] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, Charles J. Stone: *Classification and Regression Trees*, Chapman & Hall, New York, NY, 1984
- [4] Robert E. Shapire: *A Brief Introduction to Boosting*
- [5] Thomas G. Diettrich: *Machine Learning Research: Four Current Directions*
- [6] T. Hastie, R. Tibshirani, J. Friedman: *The Elements of Statistical Learning*
- [7] P. Bühlmann, B. Yu: *Explaining Bagging*
- [8] Leo Breiman: *Population Theory For Boosting Ensembles*, The Annals of Statistics 32 (2004)
- [9] Jacek Jakubowski, Rafał Sztencel: *Wstęp do teorii prawdopodobieństwa*, Wydanie II, SCRIPT, Warszawa 2001
- [10] Patrick Billingsley: *Prawdopodobieństwo i miara*, Państwowe Wydawnictwo Naukowe, Warszawa 1987
- [11] Julian Musielak: *Wstęp do analizy funkcjonalnej*, Państwowe Wydawnictwo Naukowe, Warszawa 1976
- [12] R Development Core Team (2006). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- [13] Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0.
- [14] Esteban Alfaro Cortés and Matías Gámez Martínez y Noelia García Rubio (). adabag: Applies Adaboost.M1 and Bagging. R package version 1.0.
- [15] A. Liaw and M. Wiener (2002). Classification and Regression by randomForest. R News 2(3), 18–22.
- [16] Terry M Therneau and Beth Atkinson. R port by Brian Ripley br Ripley@stats.ox.ac.uk. (2006). rpart: Recursive Partitioning. R package version 3.1-29. S-PLUS 6.x original at <http://mayoresearch.mayo.edu/mayo/research/biostat/splusfunctions.cfm>

- [17] Friedrich Leisch and Evgenia Dimitriadou. Original data sets from various sources. (2005). mlbench: Machine Learning Benchmark Problems. R package version 1.1-0.

Iwona Głowacka
Nr albumu 174093

Warszawa, 28 września 2006

Oświadczenie

Oświadczam, że pracę magisterską pod tytułem „Metody łączenia klasyfikatorów w analizie dyskryminacyjnej”, której promotorem jest Prof. dr hab. Jacek Koronacki wykonałam samodzielnie, co poświadczam własnoręcznym podpisem.

.....

Iwona Głowacka